Rate Scalable Image Compression in the Wavelet Domain

Ruzelita Ngadiran

School of Electrical, Electronic and Computer Engineering

Newcastle University

A thesis submitted for the degree of

Doctor of Philosophy

2011

Acknowledgements

In the name of Allah, the Beneficent, the Merciful. Praise and Gratitude be to Allah for giving me strength and guidance, so that this dissertation can be finished accordingly. First, I wish to express my gratitude to my supervisor, Professor Said Boussakta, for his continuous advice, support and encouragement throughout this research. Also, thanks to co-supervisor Professor Bayan Sharif for his advice and guidance. I am also thankful to Professor Ahmed Bouridane and Dr Fouad Khelifi who gave me very useful feedback on my research work. And I would like to acknowledge the financial support of the University of Malaysia Perlis and the Ministry of Higher Education, Malaysia, for making it possible for me to study at Newcastle University, United Kingdom. I also would like to thank my colleagues in the Signal and Communication research group for the friendly environment and support. I also would like to acknowledge the support and facilities from the School of Electrical, Electronic and Communication Engineering, Newcastle University. My deepest gratitude goes to my family and friends for their unconditional support and encouragement, to my husband, Khairil Aznan, and daughter, Khayla Rasyiqah, for their immense patience and understanding and most importantly to my dearest mother, Sariam Othman, to whom I owe all my achievements.

Abstract

This thesis explores image compression in the wavelet transform domain. This thesis considers progressive compression based on bit plane coding. The first part of the thesis investigates the scalar quantisation technique for multidimensional images such as colour and multispectral image. Embedded coders such as SPIHT and SPECK are known to be very simple and efficient algorithms for compression in the wavelet domain. However, these algorithms require the use of lists to keep track of partitioning processes, and such lists involve high memory requirement during the encoding process. A listless approach has been proposed for multispectral image compression in order to reduce the working memory required. The earlier listless coders are extended into three dimensional coder so that redundancy in the spectral domain can be exploited. Listless implementation requires a fixed memory of 4 bits per pixel to represent the state of each transformed coefficient. The state is updated during coding based on test of significance. Spectral redundancies are exploited to improve the performance of the coder by modifying its scanning rules and the initial marker/state. For colour images, this is done by conducting a joint the significant test for the chrominance planes. In this way, the similarities between the chrominance planes can be exploited during the coding process. Fixed memory listless methods that exploit spectral redundancies enable efficient coding while maintaining rate scalability and progressive transmission. The second part of the thesis addresses image compression using directional filters in the wavelet domain. A directional filter is expected to improve the retention of edge and curve information during compression. Current implementations of hybrid wavelet and directional (HWD) filters improve the contour representation of compressed images, but suffer from the pseudo-Gibbs phenomenon in the smooth regions of the images. A different approach to directional filters in the wavelet transforms is proposed to remove such artifacts while maintaining the ability to preserve contours and texture. Implementation with grayscale images shows improvements in terms of distortion rates and the structural similarity, especially in images with contours. The proposed transform manages to preserve the directional capability without pseudo-Gibbs artifacts and at the same time reduces the complexity of wavelet transform with directional filter. Further investigation to colour images shows the transform able to preserve texture and curve.

Declaration

I declare that this thesis is my own work and it has not been previously submitted, either by me or by anyone else, for a degree or diploma at any educational institute, school or university. To the best of my knowledge, this thesis does not contain any previously published work, except where another person's work used has been cited and included in the list of references.

Ruzelita Ngadiran

Contents

List of Figures

List of Tables

Acronyms

3D three dimensional.

bpp bits per pixel.

DCT discrete cosine transform.

DFB directional filter bank.

DWT discrete wavelet transform.

EBCOT embedded block coding with optimized truncation.

EZW embedded zerotree wavelet.

HWD hybrid wavelet and directional filter banks.

JPEG Joint Photographic Experts Group.

KLT Karhunen Love transform.

LIP list of insignificant pixel.

LIS list of insignificant set.

LsK listless specK.

LSP list of significant pixel.

LZC listless zerotree coder.

MSB most significant bit.

MSE mean squared error.

NLA non-linear approximation.

NLS no-list SPIHT.

PSNR peak signal-to-noise ratio.

SPECK set partitioned embedded block coder.

SPIHT set partitioning in hierarchical trees.

SSIM structural similarity.

WBChh wavelet-based contourlet in high frequency domain.

WBCT wavelet-based contourlet transform.

Chapter 1

Introduction

The digital representation of images and videos allows processing and archiving tasks to be integrated in multimedia platforms, computing and communications. The increasing demand for multimedia content such as digital images and video has led to great interest in research into compression techniques. The development of higher quality and less expensive image acquisition devices has produced steady increases in both image size and resolution, and a greater consequent for the design of efficient compression systems. Although storage capacity and transfer bandwidth has grown accordingly in recent years, many applications still require compression.

In general, this thesis investigates still image compression in the transform domain. Multidimensional, multispectral and volumetric digital images are the main topics for analysis. The main objective is to design a compression system suitable for processing, storage and transmission, as well as providing acceptable computational complexity suitable for practical implementation. The basic rule of compression is to reduce the numbers of bits needed to represent an image. The motivation for this study is discussed in detail in the next section, followed by a short introduction to image compression. Following this, the detailed objectives of the thesis are presented. An overview of this thesis is given in the final section of this chapter.

1.1 Motivation for the study

Digital image compression algorithms exploit the redundancy in an image so that it can be represented using a smaller number of bits while still maintaining acceptable visual quality. Factors related to the need for image compression include:

- The large storage requirements for multimedia data
- Low power devices such as handheld phones have small storage capacity
- Network bandwidths currently available for transmission
- The effect of computational complexity on practical implementation

Recent progress in image compression research offers various solutions to reduce storage requirements based on a wide range of techniques, including predictive coding, transform coding, block truncation coding, subband coding and hierarchical coding. Wavelet-based compression allows the compression parameters to be changed at the time of decoding. This is also known as rate scalability, which refers to the capability of decoding a compressed sequence at different rates. Rate scalability is useful for the transmission of compressed data between different devices. However, the challenges faced in image and video compression research are not limited to absolute storage and bandwidth concerns. An acceptable level of computational complexity in coding and decoding is also important for practical applications. As image resolution grows, the memory requirements for the compression algorithms might also grow in such a way that would limit the usage of compression techniques in embedded applications such as high resolution printers and scanners. The memory needed will be higher when using composite techniques for multispectral images. So, lower and fixed memory requirements during coding would be a good solution if the rate scalable properties of multispectral images could be maintained. This thesis investigates these problems, considering multispectral properties such as colour and satellite images and three dimensional medical images.

The efficient representation of visual information is at the centre of image compression systems. The efficiency of a representation refers to the capture of significant information about an object of interest in a smaller description. Recent research suggest that separable wavelet transform is not efficient in representing

singularities such as contours and edges in images. However, traditional separable wavelet transform benefits from available coding techniques that are efficiently able to capture the significant coefficient. This thesis investigates available solutions for directional coding in wavelet domain and their implementation for colour image compression.

1.2 Overview image compression systems

This section provides a brief overview of digital images and compression systems. The basic framework of image compression is discussed; although further details regarding image transforms/representations and coding are presented in Chapter 2. A digital image represents a two-dimensional array of samples, where each sample is called a pixel. Precision is determined by how many levels of intensity can be represented, and this is expressed as the number of bits per pixel (bpp). The value of bpp reflects different components of the colour systems used. For example, in greyscale images the values represent brightness or luminance resolution and range from 1,2,4,8,12 or 16 bpp. For RGB colour images, the values represent the intensity of each colour space, and resolution is usually 24 bpp. An ideal image compression would remove redundant and irrelevant information before the coding process. Redundancy in images can be classified as statistical redundancy or psychovisual redundancy[1]. Statistical redundancy can be classified into three types[2] :

- Spatial, due to the correlation between neighbouring pixels in an image;
- Spectral, from orrelation between colour planes or spectral bands;
- Temporal, in terms of correlation between neighbouring frames in a sequence of images.

Irrelevant information or psycho-visual redundancy refer to the limitations of or variations in the human visual system (HVS) in responding to certain stimuli under certain viewing conditions. In image compression systems, different colour components are often compressed separately as different greyscale images, and they can be represented with different spatial resolutions. However, the implementation of composite methods can further exploit spectral redundancy and

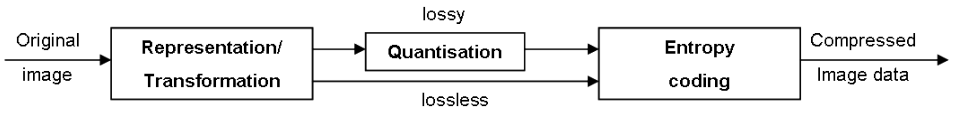


Figure 1.1: A general image compression framework

introduce rate scalable colour image coding. Image compression can be classified as lossless and lossy compression. The basic framework of both types of compression system is shown in Figure 1.1 [2]. Lossless compression is bit preserving compression, where the reconstructed image is numerically identical to the original image. This type of compression is important for applications such as medical and satellites imaging, where distortion or loss of information is unacceptable. However, lossless compression can only achieve a modest degree of compression at ratios of around $2 \backsim 5 : 1$ with a completely reversible process. Lossless algorithms usually compress the source to bit-rates close to its entropy. The quantisation process is removed in lossless compression so that the image can be recovered exactly. However, this thesis investigates lossless compression with progressive capability, where image preview is available during transmission or download, thus quantisation process is applied. On the other hand, lossy compression allows information loss so that images can be represented with reduced bit-rates, which enables higher compression ratios. This type of compression is suitable for transmission over limited bandwidths to different platforms. In general, a suitable transform is required to represent the image with a reduced dynamic range and removing redundant information. After that, the coefficient generated can be efficiently coded using quantisation techniques to reduce the bits needed to represent the images. Further entropy coding will compress the generated bit before transmission or storage.

Transform based coding has become popular since the introduction of dicrete cosine transforms (DCT) to Joint Photographic Experts Group (JPEG) image compression standards. DCTs have also been incorporated into various image and video compression standards such as MPEG1/2, H.261 and H.263. A discrete cosine transform has decorrelation properties that able to packs the energy in the fewest number of coefficients. This enables many coefficients to be discarded during the quantisation process before encoding. The DCT in a JPEG is

applied in a 8×8 unit block and compressed separately. As a result, blocking artefact obvious in compressed image when the compression ratio is high due to the differences between blocks. The JPEG standard supports colour images by compressing different colour components of the YUV colour space separately and by scaling down the chrominance components spatially by a factor of two in both dimensions. Subsequently, during the 1990s, wavelet-based compression became a heavily researched topic. The distribution values for wavelet coefficients usually centre around zero with very few large coefficients, which means that almost all of the information is concentrated in a small fraction of coefficients and can be compressed efficiently. Embedded zerotree wavelet (EZW) [3] was the first algorithm developed to show the full power of wavelet-based image compression algorithm[4]. Embedded coding is a process of encoding the transform magnitudes that allows for the progressive transmission of the compressed image. Zerotrees allow for a concise encoding of the positions of the significant values resulting from the embedded coding process. Encoding using EZW is generally fast but it does not achieve the greatest degree of compression. Set partitioning in hierarchical trees (SPIHT) [5] is a more highly refined version of the EZW. Introduced by Said and Pearlman in 1996, it gives the highest peak signal-to-noise ratio (PSNR) for a given compression ratio and is the most widely used wavelet-based algorithm for image compression. It is now the basic standard of comparison for all subsequent algorithms. A more recent image compression standard known as JPEG2000 is related to wavelet-based compression. JPEG2000 uses a block-based algorithm called embedded block coding with optimized truncation (EBCOT) to encode the transform coefficients [6]. JPEG2000 is targeted at a wide range of image compression applications, including general still image coding, video coding, variable quality coding, volumetric imaging, document imaging and wireless applications. For low complexity coding option in JPEG2000, set partitioned embedded block coder (SPECK) coding that uses block-based partitioning was introduced by Islam et al[7]. SPECK was 4.6 to 15.7 times faster than VM 3.2A in encoding due to its simplicity in encoding but of reduced in term of PSNR performance of 0.48 dB for entropy-coded versions to a maximum of 0.85 dB for non-entropy-coded versions from that of VM 3.2A. SPECK was incorporated into the JPEG2000 coding framework with the name subband hierarchical block partitioning (SBHP).

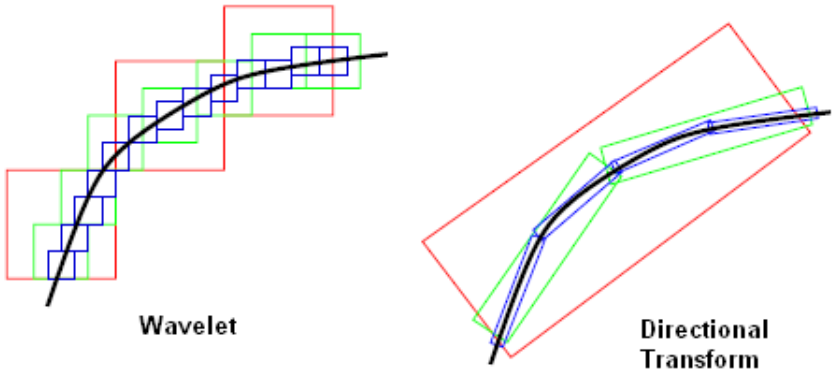


Figure 1.2: Representation of geometrical transforms against wavelet

Efficient transform or representation should be properly matched well with the quantisation method used. This has been proven through the introduction of wavelet-based transforms and the various types of zerotree and zeroblock coding that enable the significant coefficients to be captured efficiently. However, this does not eliminate the demand for more efficient image representation/transforms for compression purpose. Recent studies have shown that separable two dimensional (2D) wavelets fail to represent images optimally [8, 9]. As a result, recent research has focused on transforms with directionality. This new type of transform, also known as the geometrical image transform, aims to represent singularities in terms of smooth contours and edges in natural images (Figure 1.2[15]). Some of the leading approaches so far are the curvelet [10, 11], bandelet [12, 13, 14], contourlet [15, 9] and wavelet-based contourlet transform (WBCT) [16, 17, 18, 19]. Each approach provides directional information while maintaining some of the wavelet properties, such as the ability to provide perfect reconstruction, multiresolution representation and localised analysis.

1.3 Objectives

The main objectives of the present research can be summarised as follows :

- Review and analyse available techniques for image compression.

- Develop an efficient, rate scalable and progressive image compression technique that requires low level of working memory and has the ability to exploit redundancy in the spectral domain.
- Develop an efficient, rate scalable and progressive 3D image compression technique using zeroblock partitioning with lower and fixed working memory.
- Study and analyse the problem of singularities in the wavelet domain and available solutions to solve such problems.
- Develop a hybrid directional filter bank and a wavelet transform for compression to improve the singularities (contours or edges) representation for low bit rate image coding.
- Investigate a directional filter bank in wavelet domain for progressive colour image compression using listless embedded coding.

1.4 Thesis outline

This chapter briefly introduces the motivation for the study and the desirable features of wavelet-based coders. It also includes related work and gives an outline of this thesis.

Chapter 2 provides an overview of still image coding algorithms. The transform based image compression scheme is described and a brief review of wavelet transforms with bit plane coding are given. This includes a review and analysis of the coding mechanisms in some popular hierarchical set partition algorithm such as SPIHT and SPECK. In general, chapter 2 provides the fundamental aspects related in understanding this thesis.

Chapter 3 focused on the implementation of fixed memory listless coder to improve the efficiency of progressive colour image coding. A listless SPECK approach to colour image is proposed with the additional exploitation of chrominance similarities to improve the resulting coding.

Chapter 4 introduces the three dimensional (3D) extension to listless implementation. The aim is to reduce the memory required during 3D image coding, where zeroblock coding exploits the clustering energy in hierarchical structures of transformed images. Significant coefficients are likely to cluster together after

its 3D transforms, providing efficient compression compared to conventional 2D approach.

Chapter 5 investigates the limitations of the wavelet transform with multidimensional signals. Problems in representing curves in smooth images have been highlighted in a previous research, leading to the development of a directional filter bank to improve the transforms. A hybrid directional filter bank with wavelet transform is introduced to improve image coding performance. The transform is implemented with a listless coder for additional efficiency during coding.

Chapter 6 then investigates the implementation of the directional filter bank in wavelet domain for colour images. The proposed implementation in Chapter 5 is tested for composite colour image coding.

Finally, Chapter 7 presents the conclusions of the thesis and provides guidelines and suggestion for further investigation.

Chapter 2

Image Representation and Transform Coding

This chapter explains the process involves in image representation and transform coding techniques, with the focus on wavelet-based image compression. First, digital image representation is described, and then, the basics of image coding and the popular transforms for compression, dicrete cosine transforms (DCT) and discrete wavelet transform (DWT), are explained. A subsequent overview of wavelet-based coding techniques leads to discussion of the motivation for the present work. The information from this chapter has been collected from various sources with diverse notation conventions; the representation has been unified in order to be able to provide a consistent summary without going into unnecessary details. Performance measurement is also discussed, to prepare the ground for later experiments comparing coding algorithms and compression performance.

2.1 Image representation

A digital image is a rectangular array of dots, or picture elements arranged in m rows and n columns. The expression $m \times n$ is called the image resolution, and each dot is called a *pixel*. Bit depth is a number used to represent the value of each dot/pixel which values depends on the intensity of the images, known as bits per pixel (bpp). Based on image precision or intensity, images can be classified

into the following types [20]:

- *Binary images*, 1 bpp. Usually facsimile or black and white photographs.
- *Computer graphics*, 4 bpp. Lower precision images, often used in early generations of computers.
- *Greyscale images*, 8 or 16 bpp. Higher precision, 16 bpp usually used in medical and remote sensing images
- *Colour images*, 24 bpp. Consist of three different colour channels of primary colours for example red green blue (RGB). Can go up to 16 bits per channel for modern high resolution colour images, 48 bpp.

Colour images can also be represented in an alternative system which is also known as different colour space or device-dependent colour spaces. Some example of popular colour spaces are RGB, CIELAB, HSV and YUV. Since human visual system (HVS) are less sensitive to colour images than to luminance or brightness, RGB space has the advantage of providing equal luminance to human vision, since luminance component is present in each colour component. However, it is possible that luminance information is separated from colour information in other types of colour spaces [21]. As defined by the National Television Systems Committee (NTSC), YIQ colour spaces separate greyscale information from colour data. This enables the same signal to be used for black and white settings. YIQ component are luminance(Y), hue (I) and saturation (Q). Greyscale information is expressed as luminance (Y), and colour information as chrominance, which is both hue (I) and saturation (Q). YIQ/NTSC colour space is used in television systems.
YCbCr is another colour space that has widely been used for digital video. Here, similar to YIQ, luminance information is stored as a single component (Y), and chrominance information is stored as two colour-difference components (Cb and Cr). Cb represents the difference between the blue component and a reference value, whereas Cr represents the difference between the red component and a reference value. Another type of colour space is CMYK which is used in colour printers. The primaries in this colour space are cyan (C), magenta (M), yellow (Y) and black (K). Resolution in an image refers to the capability to represent the finer details. Higher resolutions require more complex imaging systems to represent the images in real time. Various image structures are shown in Figure

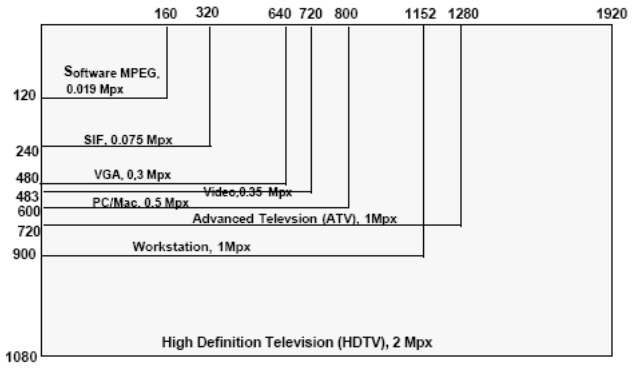


Figure 2.1: Various image structure

2.1 [20]. A direct representation of the image would require a vast amount of data, for example, an image at a resolution of 512×512 that consisting of 24 pixels occupies 786,432 bytes. At a resolution of 1024×1024 it becomes four times as big, requiring 3,145,728 bytes. The larger storage requirements of high-definition television images of a resolution of 1280×720 at 60 frames per second amounts to more than 1250 megabits per second (Mbit/s). So, the direct transmission of these video images without any compression through today's communication channels in real-time would be a difficult proposition. Even if there was enough storage available, real-time playback would be very difficult due to insufficient speed of storage devices. Compression makes the storage and transmission of multimedia content feasible. Efficient compression has therefore becomes very important due to the exponential availability of various multidimensional multimedia data, as a result of advances in image acquisition technology such as scanning and imaging technology like cameras. Although the computing performance in terms of speed and storage has increased, bottlenecks still occur in existing transmission channels.

2.2 Transform based compression framework

The definition of the transform in mathematics is to change a mathematical quantity (a number, a vector, a function, or anything else) into another form, where it may look unfamiliar but may have useful properties. The transformed

quantity is used to solve a problem or to perform a calculation, and the result is then transformed back to the original form. The useful properties of transform for compression are decorrelation and energy compaction, leaving only a small number of significant coefficients. Transforms by themselves do not provide any compression. However, by reallocating the energy of original data in compact, transforms provide the possibilities for compression. Techniques in quantisation and entropy coding applied to transform coefficients have resulted in a significant reductions in bit rates.

2.2.1 Transformation

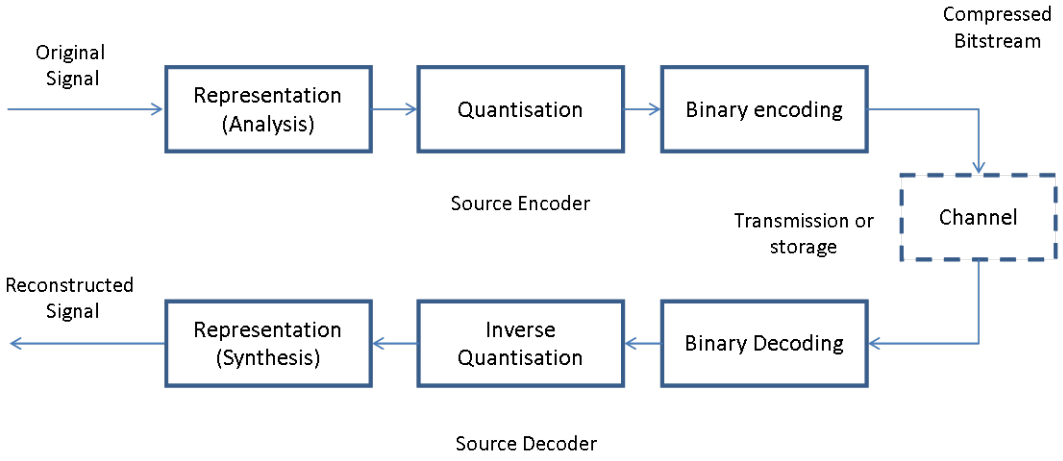


Figure 2.2: Compression and decompression

The basic operations of a transform based image compression system are represented in Figure 2.2. The main processes in encoding are signal (image) representation or transformation, quantisation and binary encoding. Binary encoding generate a compressed bit-stream suitable for transmission channel or storage. In this research, binary encoding also referred as entropy coding. In decoding, the inverse process is needed to get the reconstructed image: binary decoding, inverse quantisation and inverse transform/representation.
Redundancies in images exist in terms of statistical redundancy and irrelevancy. The transformation process will remove or suppress the redundant parts of the data in order to generate new coefficients that are independent or uncorrelated.

In this way, scalar coding can be more efficient in the transform domain than in the original signal space. In the transform domain, the information is mixed and contained in different samples so that each transform coefficient contains part of the information of all of the original pixels. The original signal or image can then be reconstructed even without all the coefficients. The transform domain can be represented by the equation below [22], where f_n, the signal, refers to an image:

$$f_n \xrightarrow[transform]{forward} a\left[m\left\langle f, \psi_m \right\rangle\right] \in \mathbb{R} \tag{2.1}$$

A forward transform applied the signal with ψ_m basis to the signal, where discrete basis is represented by $\langle f, \psi_m \rangle$ in a space of real domain, $\mathbb{R}$. Two of the most popular kinds of transform used for compression are the DCT and DWT. DCT became a popular approach for image compression due to the simplicity and low complexity implementation, and it was adopted in the Joint Photographic Experts Group (JPEG) compression standard. Despite its coding efficiency, DCT is well known for its limitations with block based algorithm that generates blocking artefacts especially at higher compression ratios. The discrete wavelet transform was later introduced to provide an alternative solution to these problems at the cost of higher complexity. The DWT algorithm is typically more memory intensive and time consuming compared to a DCT based coder like JPEG. Despite this, DWT offers benefits such as :

- Allowing image multiresolution representation
- Allowing progressive transmission / rate scalability
- Higher efficiency in term of quality of compressed image and compression ratio.

Further details on both types of transform are given in Sections 2.3 and 2.4. These sections aim to summarise the concepts that are necessary for understanding the structure of transformed coefficients.

2.2.2 Quantisation

The quantisation process follows transformation. The most suitable quantisation process depends on the choice of transform used. A quantiser is a non-linear

device that chooses representative values for ranges of input data coming from the transform, either one at a time, which is called *scalar quantisation*, or several at a time, called *vector quantisation* [23]. Quantisation is the fundamental step in achieving lossy compression. It reduces the magnitude of coefficients or rounds them to the nearest integer so that fewer bits are required to represent the image. Image frequencies are important here, because low frequencies correspond to important image features, whereas high frequencies correspond to details of the image which are less important. Thus, when a transform isolates the various image frequencies, pixels that correspond to high frequencies can be quantised heavily, whereas pixels that correspond to low frequencies should be quantised lightly or not at all. This is how a transform can compress an image very effectively by losing information, but only information associated with less important image details [24]. For approximation process during encoding, quantisation process corresponds to rounding the coefficients to an integer using a stepsize $T > 0$ as represented in Equation 2.2 [22]:

$$q[m] = Q_T\left(a[m]\right) \in \mathbb{Z} \text{ where } Q_T\left(x\right) sign(x) \left\lfloor \frac{|x|}{T} \right\rfloor \tag{2.2}$$

where $q[m]$ represent the quantized coefficients in subset of integers, $\mathbb{Z}$. This quantiser has a two times larger zero bin, so that coefficients in $[-T, T]$ are set to zero. The quantiser not only set to zero small coefficients that are smaller than T in magnitude, but it also modifies larger coefficients by rounding them. Then the resulting integer values are stored in a binary file of length R, which corresponds to a number of bits. The bits are then passed to different approach for transformation from integer into bits. For wavelet based transforms, this is the step where embedded coding occur. The aim of the process is to reduce the number of bits, R as much as possible. Further details regarding embedded coding can be found in Section 2.5. For decoding, dequantisation is the reverse process of retrieving the quantised coefficients. Equation 2.3 shows the values retrieved from quantisation at the center of the quantisation bins.

$$\hat{a}\left[m\right] = sign\left(q\left[m\right]\right)\left(\left|q\left[m\right]\right| + \frac{1}{2}\right)T \tag{2.3}$$

Based on the quantised bits, f_R the reconstructed image, using R bits can be represented as:

$$f_R = \sum_{m \in I_T} \hat{a}\,[m]\,\psi_m = \sum_{m \in I_T} Q_T\left(\langle f, \psi_m \rangle\right)\psi_m \tag{2.4}$$

where ψ_m is the orthogonal basis and the image are generated in linear domain, I_T following the quantization process, Q_T. This produces a decompression error $|f - f_R|$ which is comparable to non-linear approximation(NLA) and is the first indicator used to test the suitability of any transform for compression. More discussion on NLA is given in Section 5.4.1.

2.2.3 Entropy coding

The next step is binary encoding/entropy coding, which is lossless and reduces the number of bits further based on the statistical redundancy of the quantised value. Two of the most common entropy encoding techniques are Huffman coding and arithmetic coding. One of the main types of entropy coding creates and assigns a unique prefix code to each unique symbol that occurs in the input. These entropy encoders then compress data by replacing each fixed-length input symbol with the corresponding variable-length prefix codeword. The length of each codeword is approximately proportional to the negative logarithm of the probability. According to Shannon's source coding theorem, the optimal code length for a symbol is $-logbP$, where b is the number of symbols used to make output codes and P is the probability of the input symbol. Therefore, the most common symbols use the shortest codes. Entropy coding can be measured/estimated using the equation below:

$$E = -\sum_{i=1}^{n} p\left(x_i\right) log_2 p\left(x_i\right); \tag{2.5}$$

where E is expected value, i is the information content of x and p denotes probability mass function of x.

2.2.4 Compression performance measurement

The performance of the transform coder is evaluated by the quality of the reconstructed image. The most common measures used in the literature for lossy compression are mean squared error (MSE) and peak signal-to-noise ratio (PSNR), which are also be known as the error and logarithmic measures or rate distortion performance. For lossless compression, performance is based on the compression ratio and final bit rate. Another more recent measure, structural similarity (SSIM) is also used in the present study.

Error and logarithmic measurement

MSE is defined as:

$$MSE = \frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{i=1}^{N_y} \Big(C(x,y) - \hat{C}(x,y) \Big)^2 \tag{2.6}$$

where $\hat{C}$ is the reconstructed image with size $N_x \times N_y$ and C is the original image. The rate distortion measure or PSNR is expressed in decibels (dB):

$$PSNR = 10log\frac{255^2}{MSE} \tag{2.7}$$

where for an 8 bit greyscale image $2^8 - 1 = 255$ is the maximum value, or peak, of the signal. PSNR is a logarithmic measure, so a high value of PSNR indicates low distortion. It is worth noting that PSNR has limitations when comparing different compression systems and techniques [21].

Final bit rate

For lossless compression, the MSE value is equal to zero, thus resulting to an infinite PSNR value. The compression ratio is used as a measure of compression performance, and is based on the total bits required to reconstruct the image

numerically similar to the original image.

$$compression\,ratio = \frac{compressed\,bits}{original\,bits} \tag{2.8}$$

Structural similarity

The SSIM index introduced by Wang *et al* in 2004 [25] is a method for measuring the similarity between two images. SSIM is based on the assumption that human visual perception is highly adapted for extracting structural information [25]. The three components measured in SSIM are luminance similarity, contrast similarity and structural similarity, and the combination of these gives the value of SSIM. The SSIM metric is calculated using various windows of an image. The measure between two windows x and y of common size $N \times N$ is:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{2.9}$$

where
μ_x is the average of x;
μ_y is the average of y;
σ_x^2 is the variance of x;
σ_y^2 is the variance of y;
σ_{xy} is the covariance of x and y;
$c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ are two variables to stabilise the division with weak denominator;
L is the dynamic range of the pixel-values (typically this is $2^{\#bits\ per\ pixel} - 1$); and
$k_1 = 0.01$ and $k_2 = 0.03$ by default.
In order to evaluate image quality this formula is applied only to luminance. The resulting SSIM index is a decimal value between -1 and 1, and a value of 1 is only achievable in the case of two identical sets of data. Typically it is calculated on window sizes of 8×8. The window can be displaced pixel-by-pixel on the image, but the original authors propose to use only a subgroup of possible windows to reduce the complexity of the calculation [25]. The calculation of SSIM in this

study is based on the Matlab implementation [26]. For colour images, if the images are in RGB colour, the Matlab function *rgb2gray* is used to convert the images to greyscale.

2.3 Discrete cosine transform

The one dimensional dicrete cosine transforms (DCT) was first introduced in 1974 [27]. Since then, it has been extensively used in applications such as data, image/video and multimedia. Indeed it is also incorporated in various image and video compression standards such as JPEG, MPEG1/2, H.261 and H.263. DCT is a unitary transform which preserves the energy of a signal. In terms of energy packing capability, the principal component transform (also known as the Karhunen Love transform (KLT)) is optimal in the sense that it distributes the largest amount of signal energy into the direction of the eigenvector of the largest eigenvalue (the direction of largest sample variance), and the second largest amount of signal energy into the second largest eigenvector direction, and so on [4]. However, the KLT has a high computational cost. DCT, on the other hand, maintains the capability to decorrelate the image signal and it can be efficiently implemented in software and hardware thanks to the introduction of various DCT fast algorithms.
For image compression, DCT is applied to a block image $N \times N$ where N is typically 8 and the DCT type-II (forward) the formula in Equation 2.10 is applied to each row and column of the block.

$$C(k,l) = \alpha(k,l) \sum_{i=0}^{N-1} \sum_{i=0}^{N-1} f(i,j) \, cos \frac{(2i+1)\,k\pi}{2N} cos \frac{(2j+1)\,l\pi}{2N} \tag{2.10}$$

where $\alpha(k,l) = \begin{cases} \frac{1}{N} & \text{for } k,l = 0 \\ \frac{2}{N^2} & \text{for } k,l = 1,2....N-1 \end{cases}$

The result is an 8×8 transform coefficient array in which the (0,0) element (top-left) is the DC (zero-frequency) component and entries with increasing vertical and horizontal index values represent higher vertical and horizontal spatial

frequencies. So, the decorrelated signals can be easily quantised to reduce the overall DCT coefficients.

JPEG

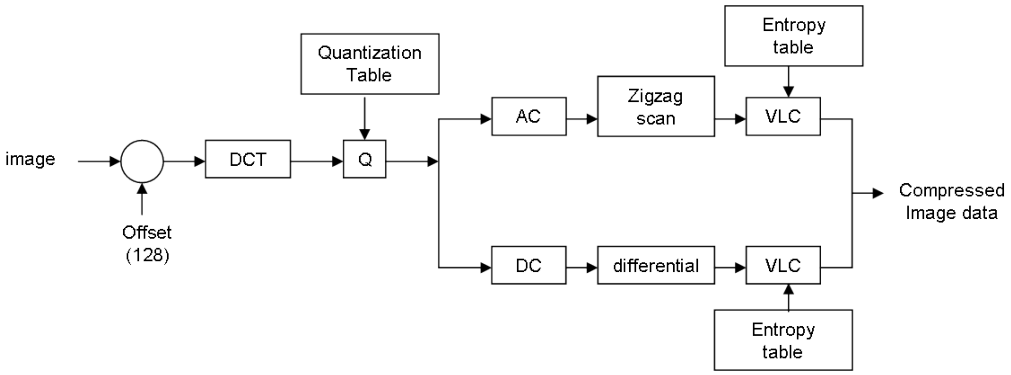


Figure 2.3: Example of DCT compression, baseline JPEG

JPEG is the ISO/IEC 10918-1 standard for the 'digital compression and coding of continuous tone still images' [29]. Figure 2.3[28] illustrates the JPEG's baseline compression algorithm, where each 8 bit sample is level shifted by subtracting $2^{8-1=7} = 128$ before being coded. This is known as DC level shifting. The 64 DCT coefficients are then uniformly quantised according to the step-size given in the application-specific quantisation matrix [28]. The quantisation matrix allows different weights to be applied according to the sensitivity of the human visual system to a coefficients of the frequency. Then, the quantised coefficients are then run length encoded (RLE) based on zigzag scan before being entropy coded using either Huffman or arithmetic coding. The clever approach of computing this transform makes the DCT extremely competitive in terms of complexity. However, the block based segmentation of the source image is a fundamental limitation of DCT based compression systems. This is also known as the blocking effect.

2.4 Discrete wavelet transform

The discrete wavelet transform provides a multiresolution representation using a set of analysing functions that are dilations and translations of a few functions. A wavelet cuts up data into different frequency components and studies each component with a resolution matched to its scale, also known as multiresolution analysis [30]. The distribution value for wavelet coefficients is usually around zero, with very few large coefficients and this means that almost all the information is concentrated in a small fraction of coefficients and therefore can be efficiently compressed. Unlike DCT-based compression, the wavelet transform operates on each frame as a whole, thus eliminating the blocking effect.
Wavelet decomposition arises from the iteration of lowpass filtering and decimation steps of a multirate filterbank. In dyadic decomposition, the iteration is on the lowpass output. An example is shown in Figure 2.4, where dyadic decomposition for 2D source image generates four subset in the first decomposition. Further decomposition is in lowpass domain to get the coarse approximation of the image. A finite number of iterations will lead to discrete-time multiresolution analysis

Figure 2.4: Example of dyadic decomposition

with a lowpass frequency response $\prod_{k=1}^{n} H_0\left(\frac{\omega}{2^k}\right)$. If the lowpass filter, h_0 satisfies the orthonormality constraint of $\sum_k h_0[k] = \frac{1}{\sqrt{2}}$, and has a vanishing moment $\sum_k k h_0[k] = 0$, then the infinite product, $lim_{n\to\infty} \prod_{k=1}^{n} H_0\left(\frac{\omega}{2^k}\right)$ converges to a function $\phi(\omega)$, whose inverse Fourier transform is the continuous time function $\phi(t)$ called the scaling function. The scaling function $\phi(t)$ is the solution to the dilation equation:

$$\phi(t) = 2\sum_k h_0[k]\,\phi(2t-k) \tag{2.11}$$

and it is orthogonal to its integer translates. The scaling function determines the wavelet $\omega(t)$ by means of the highpass filter h_1:

$$\omega(t) = 2\sum_k h_1[k]\,\phi(2t-k) \tag{2.12}$$

The set of dilates and translates $\{\omega\left(2^k t-1\right)\}_{k,l\in\mathbb{Z}}$ forms a tight frame for $L^2(\mathbb{R})$. The functional relations between Equations 2.11 and 2.12 introduce a new relationship between discrete and continuous time signal processing [4]. The span of integer translates to the scaling function $\psi(t)$ and the lowpass space V_0, where any continuous-time function,$f(t)$ can be expanded as a linear combination, $f(t) = \sum_n v_n^0 \psi(t-n)$. Here, the superscript ^{0} denotes an expansion at scale 0. $f(t)$ is completely described by the sequence $\{v_n^0\}$. So, its coarse approximation is computed with the low-pass filter of the wavelet filterbank:

$$v_n^1 = \left(\left(v^0 * h_0\right) \downarrow 2\right)[n] \tag{2.13}$$

This is implemented as low-pass filtering followed by downsampling in the two-channel structure. The details are computed with the highpass filter $h_1[n]$. Hence for a discrete sequence,v_n to be coefficients of a signal $f(t)$ at some fixed scale, the discrete wavelet transform of v_n will decompose the underlying signal f into coarse scale components and details at several intermediate scales as shown in Figure 2.5:

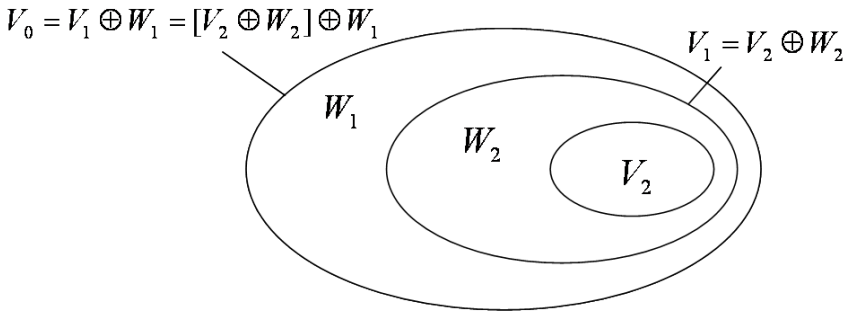


Figure 2.5: Multiresolution space representation

$$\begin{aligned} V_0 = V_1 \oplus W_1 &= [V_2 \oplus W_2] \oplus W_1 \\ &= [[V_3 \oplus W_3] \oplus W_2] \oplus W_1 = .. = V_J \oplus \sum_{j=1}^{J} W_j \end{aligned} \quad (2.14)$$

In short, the signal is represented in terms of its coarse approximation at scale J (with basis function $\psi\left(2^J t - n\right)$), and the J details (with basis function $\omega\left(2^j t - n\right), 1 \leq j \leq J$). This representation matches the multiresolution models of human and computer vision and lead to high quality image compression. This structure also enables multi-scale access to information for applications such as image browsing and the selective decoding of individual channels in multi-carrier systems [4].

2.4.1 Wavelet bases

Compression, applications such as noise removal, or high-speed calculations exploit the wavelet's ability to efficiently approximate the function (signal) with few non-zero coefficients [8]. Therefore, it is crucial to choose the wavelet bases, ψ which produce the maximum number of wavelet coefficients, $(f, \psi_{j,n})$ that are closest to zero. Three properties are key in choosing wavelet bases [8]:

- Regularity: The regularity of wavelet bases, ψ has mostly a cosmetic influence on the error introduced by thresholding or quantizing the wavelet coefficients. If ψ is smooth, then generated error is a smooth error. For image coding applications, a smooth error is often less visible than an irregular error, even though they have the same energy. Better-quality images are obtained with wavelets that are continuously differentiable from with the discontinuous Haar wavelet. Wavelet regularity increases with the number of vanishing moments. Mallat, [8] emphasizes that the number of vanishing moments and the regularity of orthogonal wavelets are related but it is the number of vanishing moments and not the regularity that affects the amplitude of the wavelet coefficients at fine scales [8].
- Number of vanishing moments: this affects the amplitude of the wavelet coefficients at fine scale. A wavelet has m vanishing moments if and only if its scaling function can generate polynomials of degree smaller than or equal to m.

- Size of wavelet bases support: these need to be reduced to minimise the number of high amplitude coefficients.

Other properties related to wavelet bases are [8, 30]:

- Orthogonality, which can be defined as an attribute where the inner product of the bases equals zero, which holds for higher dimensions as well. Orthonormal bases are both orthogonal and normalise.
- Symmetry/anti symmetry wavelet is required to avoid creating large-amplitude coefficients at the image border. Biorthogonal wavelet bases that are nearly orthogonal can be constructed with symmetric or antisymmetric wavelets bases. Therefore,they are used more often for image compression.
- Vanishing moment: for smooth regions, wavelet coefficients are small at fine scales if the wavelet has enough vanishing moments to take advantage of the image regularity. Therefore, the choice of optimal wavelet is a trade-off between the number of vanishing moments and support size.
- Compact support: Compactly supported scaling functions and associated wavelets lead to filters whose coefficients are zero outside of a finite support range. So, a filter with a shorter support is computationally less expensive than a filter with a longer support.

Wavelet with compactly supported properties include [8]:

- An isotropic Haar transform which recursively extracts details of wavelet coefficients by performing local averages/differences along the whole axis. Haar is the simplest form of wavelet filter.
- The Daubechies filters are characterised by a maximum number of vanishing moments for a certain given support. One of the advantages of having a high number of vanishing moments for the wavelet bases is that it leads to high compressibility because the fine scale wavelet coefficients of a function would be essentially zero where the function was smooth.
- The Coiflet is more symmetrical than the Daubechies wavelet.
- Symmlets are also wavelets within a minimum size of support for a given number of vanishing moments, but they are as symmetrical as possible; as opposed to the Daubechies filters which are highly asymmetrical. The index

number specifies the number of vanishing moments and is equal to half the size of the support.

- Biorthogonal wavelets allow a greater number of degrees of freedom than orthogonal wavelets. One additional degree of freedom gives the possibility to construct symmetrical wavelet functions.

Different types of wavelet families and filters were tested to see whether the quality of image reconstruction is affected. The preliminary test shows that wavelet filters with a higher number of coefficients shows better quality compared to those with fewer within the same families. Different families also give different results, as seen in Table 2.1. This test confirms that the choice of different wavelet filters also contributes to the quality of compressed images.

Table 2.1: SNR (in dB) of generated images when using different wavelet families/filters.

Compression Ratio	10	100
Haar	29.4	19.9
Daubechies 4	31	21.8
Daubechies 8	31.4	22
Daubechies 12	31.5	22
Coiflet	31.2416	22.1161
Symmlet 4	31.7094	22.4627
Symmlet 10	31.8165	22.4975

Biorthogonal wavelet bases

The term biorthogonal is used to denote the case where the analysis filter set $\{h_0, h_1\}$ is different from the synthesis filter set $\{g_0, g_1\}$. Generally, the extra freedom in the design of biorthogonal filters results in a more accurate design for low-pass filters combined with perfect reconstruction. One of the advantages of biorthogonal over orthonormal base filters is that they can both be symmetrical. On the other hand, departures from orthogonality generally have a negative impact on coding efficiency. Research by M. Barlaud [31] found that biorthogonal bases close to an orthonormal bases are suitable for image coding, so the best

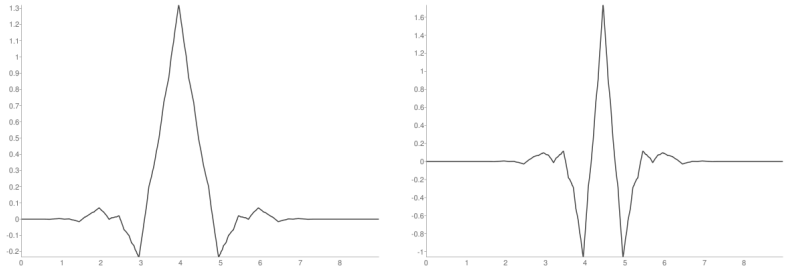
biorthogonal wavelet filters for image coding are usually nearly orthogonal [23]. The wavelet biorthogonal filter known as 9/7 and 5/3 are used for image compression in JPEG2000 and often used in wavelet image-processing applications, and are also known as Cohen-Daubechies-Feauveau 9/7 and Cohen-Daubechies-Feauveau 5/3 [31]. The 5/3 biorthogonal wavelet has $p1 = p2 = 2$ vanishing moments while the 9/7 wavelet has $p1 = p2 = 4$ vanishing moments. A filter-bank with filter sizes 7 and 9 can have 6 and 2 vanishing moments when using trivial factorization, or 4 and 4 vanishing moments as is the case for the JPEG2000 wavelet. The same wavelet may therefore be referred to as "CDF 9/7" (based on filter sizes) or "biorthogonal 4.4" (based on vanishing moments). CDF 9/7 is a non reversible wavelet transform which is suitable for lossy compression. An analysis of filter coefficients for the CDF 9/7 and 5/3 filters, which are used for dyadic decomposition, are given in Table 2.2[33]. CDF 5/3 is a reversible wavelet transform and is used for lossless compression which requires no data to be lost from rounding, and so it uses only rational filter coefficients. Figure 2.6 shows both the decomposition and reconstruction of wavelet and scaling functions for the CDF 9/7 filter [32] [32].

Table 2.2: Two sets of linear phase, biorthogonal wavelet filter coefficients

Filter Index	9/7 Filter Coefficients		5/3 Filter Coefficients	
	h_0	g_0	h_0	g_0
0	0.852699	0.788486	1.060660	0.707107
-1,1	0.377402	0.418092	0.353553	0.353553
-2,2	0.110624	-0.040689	-0.176777	
-3,3	0.023849	-0.064539		
-4,4	0.037828			

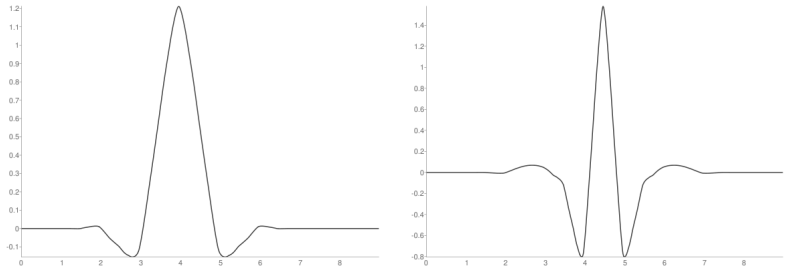
2.4.2 Multidimensional image compression

Multidimensional image sources can be images with multispectral frequency such as satellite images, volumetric images such as medical images, or video sources where the time domain is the third dimension. Colour images are the simplest form of multispectral images and are usually referred to as multicomponent. For



(a) Decomposition scaling function

(b) Decomposition wavelet function



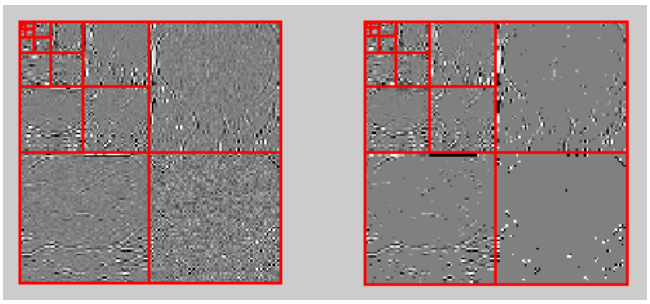
(c) Reconstruction scaling function

(d) Reconstruction wavelet function

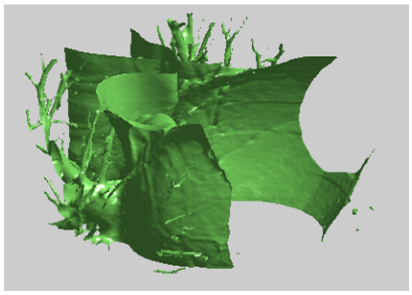
Figure 2.6: Wavelet and scaling function biorthogonal 4.4

a continuous-tone (natural) image, the principle of image compression implies that adjacent pixels have similar, although perhaps not identical, colours. However, similar colours do not mean similar pixel values [24] and therefore the colour transform should provide efficient representation. An important feature of this approach is to use a luminance-chrominance colour representation discussed in Section 2.1 instead of the more common RGB. The advantage of luminance-chrominance colour representation is that the eye is sensitive to small changes in luminance but not in chrominance. This allows the loss of considerable data for the chrominance components, while making it possible to decode the image without a significant visible loss of quality.

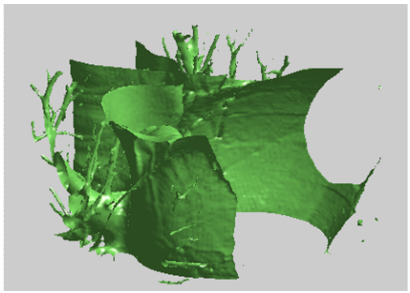
For volumetric image like medical images, direct three dimensional (3D) transform would benefit to frame decorrelation. So the redundancy between frames can be exploited during quantisation. An example of a 3D transform on Matlab using the wavelet toolbox developed by Gabriel Peyre [34] is shown in Figure 2.7. The reconstructed volume is generated after the simple thresholding of wavelet



(a) Slice of 5 levels of DWT using Daubechies 8 and its coefficients after linear thresholding



(b) Original



(c) Reconstructed

Figure 2.7: 3D compression example

coefficients (linear thresholding). For multispectral images, the 3D transform would decorrelate spectral redundancy to enable further exploitation during the compression process.

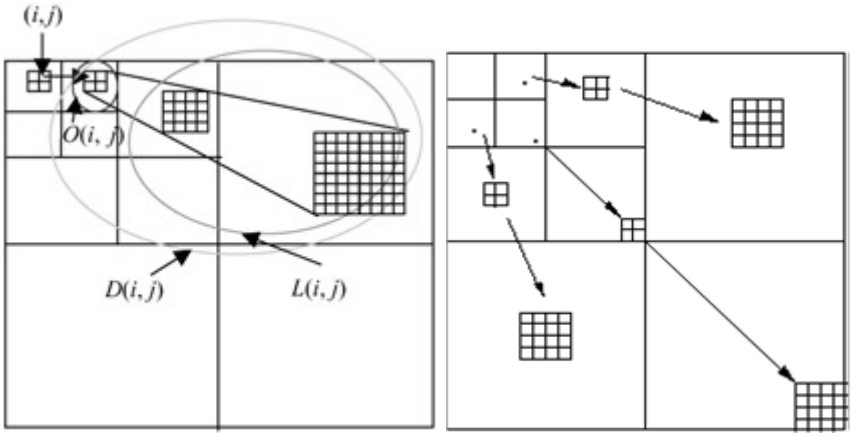
2.5 Scalable wavelet coding

The scalable coding of still images means the ability to achieve the coding of more than one quality or resolution simultaneously. Scalable coding involves generating a coded representation or bit-stream in a manner which facilitates the derivation of images of more than one quality or resolution. Bit-stream scalability is a property that allows the decoding of appropriate subsets of a bit-stream to generate complete pictures at levels of quality and/or resolution commensurate with the proportion decoded. Decoders of different complexities (from low performance to high performance) can coexist for a scalable bit-stream. While low performance

decoders may decode only small portions of the bit-stream, producing only basic quality, high performance decoders may decode much more and produce significantly higher quality. The most important types of scalability are signal-to-noise ratio (SNR) scalability and spatial or resolution scalability [35].
Progressive or scalability coding in wavelet based compression lies in set partition coding. The basic principle of set partition coding is to collect the location information of a large group of samples/pixels with maximum values below a certain low threshold [36]. The storage requirements for samples of data depend on their number of possible values which is called alphabet size. In the case of image coding, alphabet size usually requires a number of bits per sample no less than the base 2 logarithm of the number of possible integer values. For example, greyscale images require 8 bits to represent each pixel for values between 0 and 255. The location information of each group together with its size and threshold determine the bit savings of every group and hence the compression compared to raw storage using a full alphabet size for each sample. Irrespective of how the location information is gathered, the potential compression gains can be accessed simply by adding the number of bits associated with the actual values of samples. This technique will be efficient in the transform domain where the source primarily contains sets of samples with small maximum values. Set partition is a class of methods that can take the properties of image transforms. The members of this class differ in the method of forming the partitions and the ways in which the location information of the partitions is represented. The first part of this thesis deals with these two important classes (methods of forming partition and representing location information) in order to apply set partition algorithms to multidimensional multimedia signals.

2.5.1 EZW

The embedded zerotree wavelet (EZW) [3] was the first algorithm to show full power wavelet based image compression. Embedded coding is the process of encoding the transform magnitudes, allowing for the progressive transmission of a compressed image. Zerotree allows for a concise encoding of the positions of



(a) SPIHT [28] (b) EZW

Figure 2.8: Refinement in SPIHT zerotree and comparison with EZW

the significant values that result from the embedded coding process. The EZW algorithm is developed in several steps. Since the wavelet transform provides a pyramid-like multiresolution representation of the image, zerotree coding then provides binary significance maps indicating the positions of the significant coefficients. This is followed by the process of ordering the coefficients that are deemed significant. Figure 2.8(b) shows the zerotree relation exploited in EZW. It should be noted that each root node in the top subband has three offspring, one in each high frequency subband at the same decomposition level, and all other coefficients have four children in the lower decomposition subband of the same orientation. Encoding using EZW is rapid, but does not achieve the greatest compression. Arithmetic encoding is recommended to further compress the bit-stream.

2.5.2 SPIHT

Set partitioning in hierarchical trees (SPIHT) is a highly refined version of EZW [5]. It was introduced by Said and Pearlman and gives the highest PSNR for given compression ratios. Packet transforms are selected to allow a different number of decompositions between the spatial and temporal dimensions, enabling better compression of the dyadic decompositions. The SPIHT algorithm refines the EZW algorithms organisation of the wavelet coefficients across the subbands to achieve a better coding of the significance maps (see Figure 2.8). It is the most

widely used wavelet based algorithm for image compression and has become the basic standard benchmark for comparison with all subsequent algorithms. SPIHT has been extended to the 3D area, and research into SPIHT has progressed in a wide variety of fields. SPIHT utilizes three basic concepts [5]:

- Sets are searched for in spatial orientation trees in a wavelet transform. Spatial orientation trees are groups of wavelet transform coefficients organised into trees rooted in the lowest frequency or coarsest scale subband with offspring in several generations along the same spatial orientation in the higher frequency band. Each node consists of 2×2 adjacent pixels. Each pixel in the node has 4 offspring except at the highest level of the pyramid. Zerotree is introduced to exploit self similarity and magnitude localisation properties.
- Partitioning the wavelet transform coefficients in these trees is defined by the level of the highest significant bit in a bit plane representation of their magnitudes.
- Coding and transmitting bits associated with the highest remaining bit planes first

The SPIHT process consists of two main stages; sorting and refinement. The sorting process involves a magnitude threshold of 2^n being set, where n is the level of significance.

1. If tested coefficients $\geq 2^n$, the significant coefficients is isolated in list of significant pixel (LSP) and bit '1' is sent.

2. If tested coefficients $\leq 2^n$, the insignificant coefficients is isolated in list of insignificant pixel (LIP) and bit '0' is sent.

3. All coefficients that are less than the set of insignificant coefficients is isolated to list of insignificant set (LIS) so bit '0' is sent.

The refinement process involves:

- n lowered in unit increments from $n\,(max)$. The current threshold value is lowered.

- if the n^{th} value of LSP if higher than, n, the value is emitted to code stream. The significant coefficients is sent to bit stream.
- n decremented, $n-1$, LIP is tested and the process is continued.

The SPIHT algorithm can be extended to a 3D transform by providing different sorting tree algorithms suitable for the 3D wavelet coefficients generated. The SPIHT algorithm is extended to 3D-based on the sorting of the 3D tree. A 3D node acts as a block of eight adjacent pixels with two extending into each dimension, forming a node of $2 \times 2 \times 2$ pixels. The refinement process is the same as mentioned before. In general, when used with the CDF-9/7 wavelet, SPIHT outperforms all other techniques. This is because it has a dedicated arithmetic compression scheme that exploits the redundancies in the wavelet subbands. However, this zerotree method is most suitable when the coefficients follow the decaying spectrum hypothesis [3]. Low-frequency components generally follow this trend but high frequency images do not. Thus, while we may see a higher PSNR value with SPIHT on some images, this does not necessarily mean that the image looks good, because it may have higher frequency content. A preliminary test for SPIHT is conducted using the Matlab wavelet toolbox. The rate of the compressed image is set at 1 bit per pixel. In this experiment, the differences between image qualities based on the PSNR measure of each wavelet filter were selected randomly. The results in the Table 2.3 verify the importance of selected

Table 2.3: SPIHT test using various filters

Type of wavelet filter	PSNR (dB)
Bior6.8	39.92
Bior4.4	39.85
Haar	37.49
Coif1	39.06
Sym1	37.49
Sym2	38.95
Db8	39.63
Dmyer	39.83
Rbio6.8	39.58

wavelet bases to the quality of the compressed image with the SPIHT implementation. From Table 2.3, the tested filters show better results on filters with more

coefficients such as Bior6.8 and Bior4.4 and lower in simpler filter such as Haar.

2.5.3 SPECK

The set partitioned embedded block coder (SPECK) [37] is a block based embedded coder which offers lower complexity than SPIHT. Like SPIHT, SPECK consist of two main processes, sorting and refinement. Figure 2.9[37] summarises the partitioning or sorting process in SPECK. The idea behind SPECK partitioning

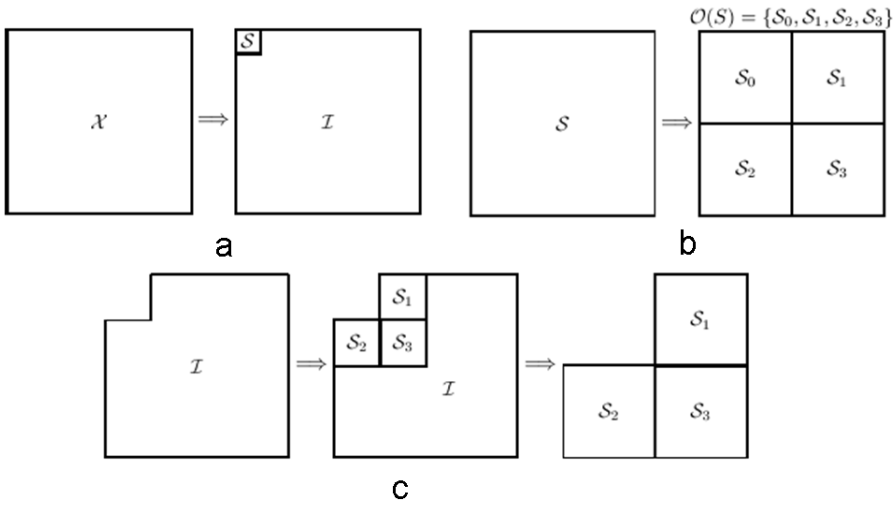


Figure 2.9: Partitioning process in SPECK

is to exploit the hierarchical pyramidal structure of the subband decomposition, where it is more likely that energy is concentrated at the uppermost levels of the pyramid, and as one goes down the pyramid, the energy content decreases gradually [37]. The term x in Figure 2.9 represents the image transformed coefficients, where S is for the lowest subband coefficients and I the insignificant set. The sorting process is based on the test of significance. If set S is found to be significant, it will be partitioned into four smaller subsets until the significant coefficient is found. SPECK reduces the complexity of SPIHT because it uses zeroblock instead of zerotree. Only two lists are required to track the coordinates in SPECK, LIS and LSP. Fewer lists and simpler partitioning processes leads to a faster encoding and decoding process in SPECK compared to SPIHT.

2.5.4 EBCOT

Embedded block coding with optimized truncation (EBCOT) [6] is a wavelet-based image compression algorithm adopted for the JPEG2000 standard. The EBCOT principle is to divide each subband into blocks (termed code-blocks) and then code each individually. The bits resulting from coding several code-blocks become a packet and the packets are the components of the bit-stream [6, 24]. The bit-stream also contain markers which can be used by the decoder to skip certain areas of the bit-stream and to reach certain points/rates quickly. A visual representation of an EBCOT code-block is shown in Figure 2.10[8]. With

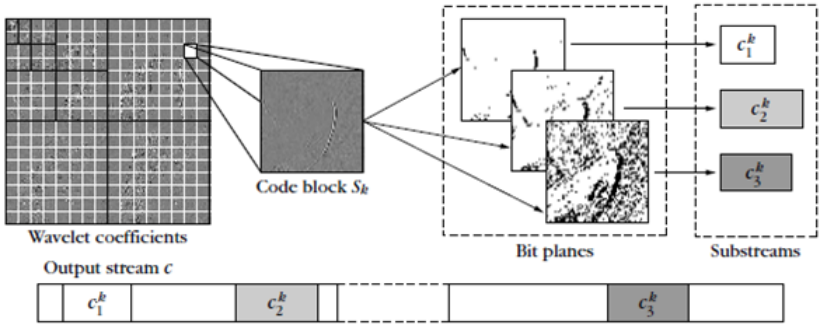


Figure 2.10: JPEG2000 compression using EBCOT

markers, EBCOT coding can display certain regions of the image before other regions, a properties known as random access capabilities or Region of Interest (ROI). These makers are also used to progressively decode (rate scalable) the image in one of several ways. The EBCOT bit-stream is organized in layers, where each layer contains higher-resolution image information which enable resolution scalability. Thus, decoding the image layer by layer is a natural way to achieve progressive image transmission and decompression [24]. Its use of layered block coding, fractional bit planes, block-based rate distortion (R-D) optimization, and context-based arithmetic coding means that EBCOT technique is able to generate a feature rich bit-stream which is rate scalable, resolution scalable and with random access, but at the cost of high complexity. In this thesis, the state of the art JPEG2000 standard that uses EBCOT is tested using the Kakadu Software developed by Taubman [38]. Summary of the scalable coding algorithms described previously is in Table 2.4.

Table 2.4: Summary of scalable coding algorithms

Technique	Bit partitioning	Complexity
EZW	Zerotree	Low
SPIHT	Zerotree	High
SPECK	Zeroblock	Low
EBCOT	Block Based Zerotree	High

2.6 Summary

A significant breakthrough for wavelet based image and video coding was the discovery of zerotrees for coding two dimensional wavelet coefficients. The zerotree approach is commonly used since it achieves high performance and generates a progressive bit-stream by exploiting the inter-subband dependency among insignificant wavelet coefficients. This is achieved by reorganising the transformed coefficients in space-scale trees.

EZW was then improved with the SPIHT algorithm which exploits the spatial orientation tree to find significant coefficients. In addition to the zerotree concept, other techniques have been proposed such as SPECK which exploit intra-subband coefficients in the wavelet domain. Another wavelet based coder is EBCOT [6], which was adopted in the JPEG2000 standard [39], EBCOT combining layered block coding, R-D optimisation, and context-based arithmetic coding in an efficient and highly scalable way.

As with all other new technologies, various non-technical issues that play a part in the adoption rate of new systems in addition to objective performance, in determining commercial success. For example JPEG2000 standard struggles to generate strong industrial friction more than five years after its introduction despite the obvious technological advantages over the JPEG standard. This shows that external industrial factors, such as licensing fees, the cost of technology shifts, or increased computational complexity, are clearly as important when it comes to the adoption of the technology.

This chapter has reviewed the transform-based image compression techniques. Every aspect of compression plays an important role in improving its performance in term of compression ratio and the rate of distortion when the compressed image

reconstructed. In general, wavelet-based approaches and in particular, scalable coding algorithms are emphasized. Details of each step in transform-based compression have been explained. In particular, the wavelet transform and the state of the art coding methods SPIHT and SPECK have been discussed in detail, and these are used later as a benchmarks for the algorithm proposed in this study.

Chapter 3

Listless Implementation for Embedded Colour Image Coding

3.1 Motivation

Trivial methods of independently coding each of the resulting spectral planes with high performance wavelet based greyscale image codecs such set partitioning in hierarchical trees (SPIHT)[5], JPEG2000 or subband block hierarchical partitioning(SBHP)[7] is highly inefficient since nonlinear dependencies at high transition regions (such as edges) remain among the spectral planes [40]. This chapter proposes an efficient algorithm for composite colour image compression, with listless implementation based on the set partitioned embedded block coder (SPECK) partitioning rules. Listless coders such as no-list SPIHT(NLS) [41] and listless specK(LsK)[42] have primarily been designed for greyscale image or single component images. The objective of the present work is to develop an algorithm that exploits redundancy in colour spaces while maintaining the embedded properties and low complexity quadtree partitioning, but with reduced memory requirements so as to be suitable for hardware implementation. Colour images are first transformed to luminance chrominance (YCbCr) planes, and then a wavelet transform is applied to each plane. A reduction in the memory required is achieved with the introduction of a state marker that matches each colour plane to eliminate the list from the original colour-SPECK(CSPECK). The dynamic

memory required in CSPECK is replaced with a low memory and fixed state marker. The location of wavelet coefficients is then mapped based on scanning order or linear indexing that matches the transformed coefficients. The scanning method used is the one dimensional Z-curve, also known as Morton order that is very good at preserving spatial proximity. This one dimensional scanning method also offers computational efficiency and algorithmic simplicity. The proposed algorithm then encodes the decorrelated colour plane as one unit and generates a mixed bit-stream. The indexing and markers are modified to jointly test the chrominance planes. The resulting algorithm maintain full embeddedness and the precise control of bit rate required for image reconstruction. The performance of the proposed algorithm is tested on lossy and lossless compression of a 24 bit colour image. The performance of proposed coder is comparable to that of CSPECK, SPIHT and JPEG2000.
This chapter is organised as follows. Section 3.2 is the introduction to this chapter and followed by a background and reviews work related to colour coding, the SPECK algorithm and the attractive listless coder in Section 3.3. Section 3.4 explains in detail the algorithm for the proposed coder. This is followed by a presentation of the numerical results in Section 3.5, with performance compared with that of the CSPECK coder, JPEG2000 and SPIHT. The performance measurements include rate distortion analysis, memory used in the algorithm and coder complexity. Finally, conclusions are drawn in Section 3.6.

3.2 Introduction

The increasing demand for multimedia content such as images and video has resulted in great interest in research into compression techniques. Compression in the wavelet domain has become significant since the introduction of embedded coding algorithms such as embedded zerotree wavelet (EZW) [3] and SPIHT [5], which offer the effective reordering and coding of transformed coefficients into progressive and rate controllable bit-streams. Both algorithms use the simple prioritisation and segregation of significant coefficients into ordered bit planes. Further embedded coding technique has been introduced, which is based on a

block partitioning (zeroblock) algorithm known as SPECK. SPECK offers an approach which is less complex and provides comparable performance to SPIHT, but with faster encoding and decoding times [37] [7]. The SPECK algorithm properly and efficiently sorts sets of transform coefficients according to their maximum magnitude with respect to a sequence of declining thresholds. The sorting mechanism is based on quadtree partitioning guided by a test of significance. This allows the processing of different regions of the transformed image based on their energy content.

The basic principle of set partition coding is to collect the location information of large group of samples/pixels with maximum values below a certain low threshold [36]. Irrespective of how it is gathered, the location information for each group together with its size and threshold determines the bit savings involved. So, compression is achieved through the bit savings for every group compared to the actual bits which would be required to represent each pixel. If this technique was applied directly to raw images, the addition of location information would require more bits, so a transform is required to group together the samples with small maximum values. Image transforms can provide highly effective methods of forming partitions and representing the location information. These definitions have led to the development of state of the art coders, such as SPIHT and SPECK. SPIHT and SPECK, which gather the location information for groups of samples at certain thresholds through the introduction of lists into the algorithm. Such lists grow when the groups contains more samples if the threshold is lowered. Due to the nature of these algorithms, the list is required to be added, removed and moved, which leads to variables and data dependency, dynamic memory requirement and memory management. Listless approaches were then introduced to deal with these problems. Listless coders remove the need for location information in the list through the introduction of scanning methods to identify the location of the coefficients and sparse markers to represent the state information of each coefficient. This state information is kept in an array of fixed size of 4 bits per coefficient, in order to enable the fast scanning of set partitions.

Fixed size state information or listless coders were developed by Lin et al - the listless zerotree coder (LZC) [43, 44, 45, 46]; Wheeler - the NLS [41] and Latte et al - the LsK [42]. The LZC performs an in-depth first search of the trees, and this

algorithm was refined in the NLS that facilitates a breadth first search of zerotree. The LsK, however, uses the SPECK approach to intra-subband exploitation using simpler algorithms and maintaining the breadth of the first search approach. However, these coders can only be implemented with greyscale images. The implementation of these coders is more complex with colour images, because the independent implementation of colour channels introduces colour artefacts and rate scalable coding is not supported . Colour artefacts can be avoided by using appropriate colour spaces.

In the present study, an efficient SPECK based compression technique using quadtree partitioning and a zeroblock coder is implemented for colour images. The coder is modified to jointly encode the chrominance planes together, in order to improve coder efficiency and reduce complexity. The fixed memory required for this listless coder suggests that the algorithm is suitable for hardware implementation. The zeroblock coder also offers low complexity compared to conventional zerotree coders. Lists are eliminated and replaced by a state marker. Instead of searching the tree to find predictable significants, special markers are placed on certain nodes of the trees during initialisation. These markers are updated when new insignificants or significants are formed by the quadtree partitioning. This sparse marking enables the coefficient to be scanned efficiently and skips insignificant coefficients as the scan moves to a different subband.

3.3 Background and review

Full colour images are represented by three colour spaces: red, green and blue, called RGB space. These colour planes are usually highly correlated, so less correlated spaces such as luminance and chrominance are used for efficient compression. In this work, the colour space of luminance and chrominance, $YCbCr$ for digital images is used, which is similar as defined for JPEG2000. In this colour space, Y is the component of luminance while Cb and Cr provide the colour information. The colour information is stored as two colour difference components: Cb and Cr.

3.3.1 JPEG2000

Figure 3.1 shows a block diagram of the core encoder for colour image compression using JPEG2000. Notice that JPEG200 first applies DC offset to RGB colour components by 2^{B-1} to simplify certain processing to its coder [28]. The colour

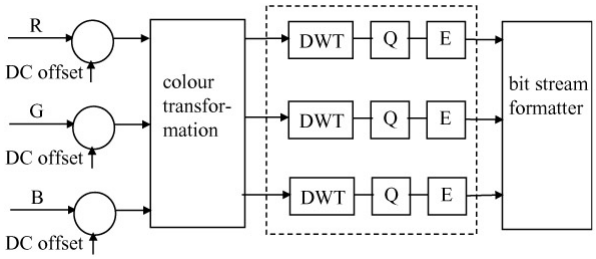


Figure 3.1: Core encoder of JPEG2000 for colour image coding

transformation from RGB to $YCbCr$ used in JPEG2000 is defined as follows [28]:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.1}$$

This colour transformation is called irreversible colour transform (ICT) as specified for lossy compression. It can only be used with the floating point wavelet transform which is irreversible. For lossless compression the reversible colour transform (RCT) is used with an integer transform which is reversible. RCT is defined as follows [28]:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.25 & 0.5 & 0.25 \\ 15 & -1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.2}$$

The colour transform defined for both lossy and lossless compression in JPEG2000 is used for all types of coders developed in the present research, and this also applies to the wavelet transform used. JPEG2000 uses a 9/7 tap filter for lossy and a 5/3 filter for lossless compression. After the wavelet transform, a quantisation process takes place, (Q in Figure 3.1), where the sixe of each quantiser step can vary from band to band. Moreover, since JPEG2000 uses image tiles which are

coded independently, a quantiser step can also vary from tile to tile. Wavelet coefficients are most efficiently coded when they are quantised by successive approximation, which is the bit plane representation of the quantised coefficients. Therefore, JPEG2000 uses embedded block coding with optimized truncation (EBCOT), coding as described in Section 2.5.4. Once the entire image is compressed, the bit-stream generated by the individual code blocks is postprocessed to facilitate various functionalities of the JPEG2000 standard (as represented by the bit-stream formatter block in Figure 3.1).

3.3.2 Colour coding in the wavelet domain

The early implementation of SPIHT with colour images used the Karhunen Love transform (KLT) to decorrelate spectral information prior to SPIHT coding [5, 47, 48]. The evaluation of this method showed that the performance of KLT is comparable to that of YUV method, but it has the disadvantages of higher computational complexity than YUV and high dependency to its source. The CEZW implements an embedded scalable colour image coder by exploiting the colour components of the YUV plane [49]. The coding strategy of CEZW is similar to that of EZW but with changes to the dominant pass. Luminance is scanned first and appropriate symbols are assigned accordingly. Then the chrominance plane is scanned alternately until all components are encoded. The rate of distortion performance in CEZW is lower than that of the SPIHT algorithm.
The CSPIHT was then introduced by Kassim and Wei [50] with modifications to the SPIHT spatial orientation tree (SOT) structure. The CSPIHT uses a partially linked SOT across different spectral planes. This method embeds both chrominance and luminance in the coded bit-stream and involves lower computational complexity than using SPIHT with KLT. CSPIHT also performs better than SPIHT+KLT, especially with a five-level decomposition wavelet. Fouad et al subsequently modified the SPIHT in [51, 52, 53, 54], providing a better peak signal-to-noise ratio (PSNR) than the original version SPIHT [48], especially colour compression at very low bit rates. The main modification consisted of grouping the trees within each chrominance plane in list of insignificant set (LIS) during the sorting pass according to the virtual parent-descendants rela-

tionships specific to the chrominance component [51]. The same authors also implemented lossless colour/multispectral image coding using SPECK by joining two correlated planes in the significance test of the blocks, resulting in a significant improvement in the final bit rate [55, 56]. Other work then took the human visual system (HVS) into account [57, 58] to improve compression performance. Recent work by Bayazit [40] on colour image compression uses all three colour planes during coding process introducing an adaptive spectral transform across the colour plane of an image. This transform improves the performance of the image compression with slight increase its coding complexity. Exploiting similarities between the correlated colour planes brings significant improvements to the rate of distortion performance, especially at low bit rates compression without adding to the complexity of the coder.

3.3.3 Reduced memory coder

There have been several attempts to reduce the memory required in the process of coding and decoding wavelet coefficients [41, 42, 43, 44, 45, 46, 57, 59, 60]. The idea of listless significance map coding was initially proposed by Lin and Burgess [43, 44, 45, 46] in their work on the LZC for colour images [43]. The concept was then further developed by Shively et al. [59] and Wheeler and Pearlman [41]. The LZC uses two significant bit maps with one bit per coefficient to store the coordinates of significant coefficients and tree information, thus making its implementation more hardware friendly. The PSNR performance of LZC is low, at about 0.51 dB, compared with that of the SPIHT. Other recent listless algorithms based on the SPIHT coder have also been implemented [60, 57]. A study by Wheeler [41] improved the performance of the algorithm by implementing a marking scheme that facilitated breadth first searching. As a result, the NLS algorithm produced bit-streams with a similar number of bits (but in a different order) and generated results that were practically identical to SPIHT. Among all of the listless implementations described so far, only the LZC has been implemented on still colour images to provide true scalable embedded coding.
The LsK is another low memory greyscale image coder that implements a structure similar to NLS but using the SPECK partitioning rule [42]. This approach

offers lower computational complexity in terms of the partitioning rule while maintaining the fixed memory approach of a listless coder. The proposed algorithm is motivated by the development of previous listless coders such as LsK, NLS and LZC, and with the efficient zeroblock partitioning rule for colour images, with the aim of reducing the complexity of the scalable colour image coder.

3.3.4 CSPECK

The CSPECK is a block based hierarchical image coding scheme that is completely embedded, employs progressive transmission, and is low in computational complexity. The SPECK outperforms SPIHT on images with high frequency content, such as Barbara, while CSPECK outperforms JPEG2000 and SPIHT for luminance components in most images [37]. In wavelet based progressive image compression, SPECK can be considered the most efficient coder compared to JPEG2000 and SPIHT, and previous reports indicate that it performs 4 to 8 times faster than JPEG2000 [37, 7]. The CSPECK treats all colour planes as one unit at the coding stage and generates a mixed bit-stream as shown in Figure 3.2 it containing the information about each colour plane denoted by Y,U,V. This maintains full embeddedness and precise control of bit rate because it can stop at any point of the bit stream to reconstruct the colour image of every plane.

YYYYYYYUUUUUUUUVVVVVVVVVVVV
YUVYUVYUVYUVYUVYUVYUVYUVYUV

Figure 3.2: Compressed colour bit-streams : conventional and embedded

The SPECK only requires two lists, unlike SPIHT which requires three. In CSPECK, one list of significant pixel (LSP) and one list of insignificant set (LIS) are used for each transformed plane, and each is initialised with the corner coordinates of its top level S as shown in Figure 3.3, which also represent the basis of the SPECK partitioning rule. Significant points among the three planes are mixed in the single LSP. When the threshold is lowered (n-1), the LIS sets of YUV are visited in turn on the three lists for SPECK sorting passes. The refinement pass takes place on a single LSP by sending n^{th} bits of the binary expansion of magnitude found significant. The procedure is then repeated until the bit budget is met.

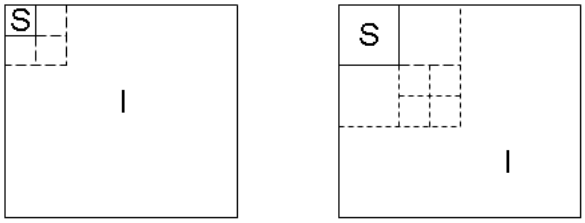


Figure 3.3: SPECK partitioning rule

3.4 The proposed coder

In the coding algorithm in this study, the LsK [42] is expanded for implementation with composite colour images. The algorithm treats all colour planes ($YCbCr$) as one unit at the coding stage and generates a mixed bit-stream. For identification, LsK for colour image is referred as Listless CSPECK (L-CSPECK). The proposed algorithm maintains listless properties with modifications to jointly encode the chrominance planes in order to exploit the similarities between them. As with the original LsK [42], the proposed algorithm consists of three stages: initialisation, sorting pass, and refinement. Originally, test of significance of blocks were conducted for each $YCbCr$ colour plane in the L-CSPECK. By combining the test of significance in the chrominance plane, the algorithm becomes more efficient. This is achieved by modifying the scanning order and initial state marker of the chrominance plane. There are three main components of listless implementation: linear indexing, the state marker and the sorting algorithm. Linear indexing is a mapping technique of the transformed coefficients to a scan order and it is applied during initialization stage. After that, the state marker is initialized to generate a set of insignificant pixels and skipped coefficients. Further details of each component are given in the following sections.

3.4.1 Linear indexing

Application of the dyadic wavelet transform results in a hierarchical ordering of wavelet coefficients (the subband pyramids). Their order is organised in such a way that the coefficient's magnitude tends to decrease with the depth of the subband. This property is exploited by a bit plane encoding scheme with a significance test, as shown in Equation (3.3).

$$S_n(B) = \begin{cases} 1 & if max_{i,j \epsilon B |C_{ij}|} \geq 2^n \\ 0 & otherwise \end{cases} \tag{3.3}$$

Linear indexing provides an efficient way of mapping the transformed coefficients to scan order during the partitioning process. The linear indexing used here is based on Morton ordering or recursive Z. It generates a hierarchical order similar to a subband pyramid as in Figure 3.4, and uses a single number to represent the index of the coefficient which can be simply generated through bit interleaving [41]. Morton ordering offers good locality-preserving behaviour and provides suitable structures for mapping multidimensional data onto one dimension. This indexing efficiently supports the treatment of coefficients with one operation, assuming the usual subband data arrangement. This particular format offers computational and organisational advantages [41, 42, 61] and facilitates the breadth first search algorithm. The straightforward approach for listless colour image is

1	2	5	6	17	18	21	22
3	4	7	8	19	20	23	24
9	10	13	14	25	26	29	30
11	12	15	16	27	28	31	32
33	34	37	38	49	50	53	54
35	36	39	40	51	52	55	56
41	42	45	46	57	58	61	62
43	44	47	48	59	60	63	64

Figure 3.4: Morton ordering for Y Plane

to apply similar ordering to each plane and enable rate scalable properties on the recontructed colour image. The scan order for L-CSPECK is shown in Figure 3.5.

The Morton ordering is generated by bit interleaving of each coordinate. Let say

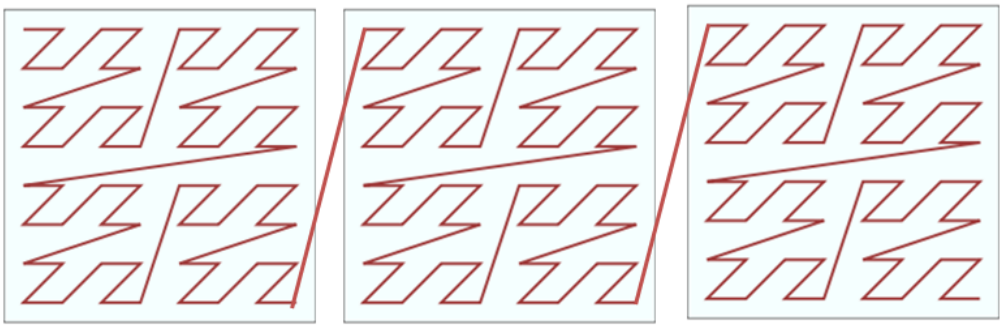


Figure 3.5: Linear indexing to $YCbCr$ plane

colour image is represented as $f(x, y, c)$, each coefficients required $log_2(x, y, c)$ bits to identify the coordinate of each dimension. For 256×256 image, 8 bits are required to represent the coordinate. $x_0 \rightarrow x_7, y_0 \rightarrow y_7$ So, x_o represents the first bit of the X axes. This bit interleaving is shown in Figure 3.6 and 3.8.

The proposed algorithm requires different scanning decisions to combine the

$$x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0$$
$$y_7, y_6, y_5, y_4, y_3, y_2, y_1, y_0$$
$$x_7y_7, x_6y_6, x_5y_5, x_4y_4, x_3y_3, x_2y_2, x_1y_1, x_0y_0$$

Figure 3.6: Bit interleaving in Y plane

chrominance significant test, in order to exploit the similarities between chrominance planes (CbCr). To do this, the scanning considers both chrominance planes in the linear index. The bit interleaving of coordinates in plane Cb and Cr, $f(x, y, c)$ is used to generate the linear index for scan order. Figure 3.8 illustrates the bit interleaving to generate the joint scanning or linear index for the chrominance planes, where c_1 and c_0 represent planes Cb and Cr respectively. An example of the generated indexing is shown in Figure 3.9. Note that this type of scanning ensures interchanging between chrominance planes during the significant tests as shown in Figure 3.7. In terms of hardware considerations, Wheeler and Settha [41, 61] have mentioned that this index conversion is trivial, but it is

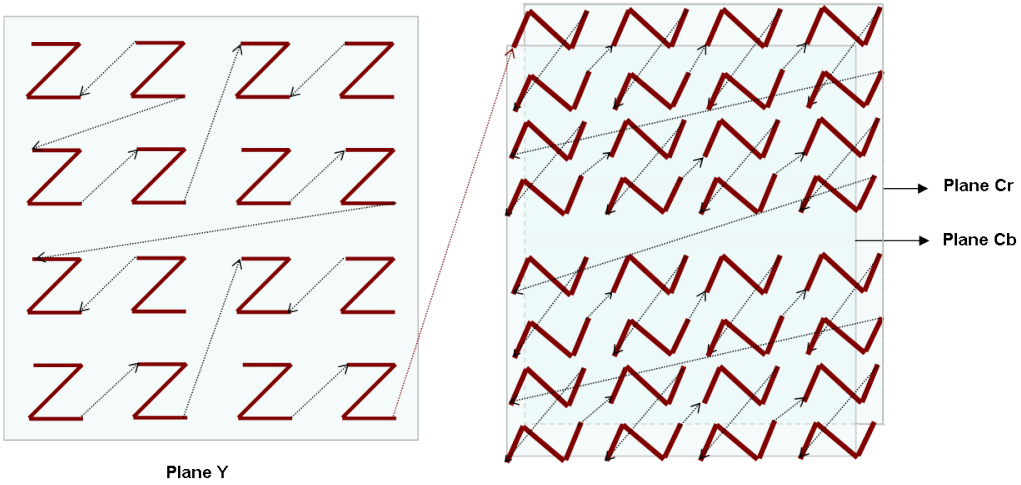


Figure 3.7: The modified scanning order

$$x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0$$
$$y_7, y_6, y_5, y_4, y_3, y_2, y_1, y_0$$
$$c_1, c_0$$
$$x_7y_7, x_6y_6, x_5y_5, x_4y_4, x_3y_3, x_2y_2, x_1y_1c_1, x_0y_0c_0$$

Figure 3.8: Bit interleaving in chrominance plane

1	3	9	11	33	35	41	43
5	7	13	15	37	39	45	47
17	19	25	27	49	51	57	59
21	23	29	31	53	55	61	63
65	67	73	75	97	99	105	107
69	71	77	79	101	103	109	111
81	83	89	91	113	115	121	123
85	87	93	95	117	119	125	127

Plane Cb

2	4	10	12	34	36	42	44
6	8	14	16	38	40	46	48
18	20	26	28	50	52	58	60
22	24	30	32	54	56	62	64
66	68	74	76	98	100	106	108
70	72	78	80	102	104	110	112
82	84	90	92	114	116	122	124
86	88	94	96	118	120	126	128

Plane Cr

Figure 3.9: Scanning order for chrominance plane

not directly supported by general purpose CPU. Consequently, for faster software conversion, this indexing/ordering can be implemented using a fixed lookup table that maps one byte to two bytes with zero bits, padded between original bits.

The conversion is simple and fast, using a bit spreader table combined with shift and bitwise OR operations.

3.4.2 State marker

The mapped coefficients (scan order) have a direct relations with the state marker. The markers listed below play an important role in keeping track of the partitioning rule. During initialisation, all coefficients are marked as null, and then a few coefficients are marked according to the estimated subband positions of the transformed coefficients. 4 bits per coefficient are required to stating the marker. Figures 3.10 and 3.11 give an example of few coefficients marked during initialisation and the difference between the marker initialised in L-CSPECK and the algorithm proposed here. Map refers to the indexing generated for scanning

$$\begin{gathered}\text{mark}(ma \times mb + 1 : ma \times mb + 64) = \text{MIP};\\ \text{mark}(ma \times mb + 8^2 + 1)\text{=MS3};\\ \text{mark}(ma \times mb + 16^2 + 1)\text{=MS4};\\ \text{mark}(2 \times ma \times mb + 1 : ma \times mb + 64) = \text{MIP};\\ \text{mark}(2 \times ma \times mb + 8^2 + 1)\text{=MS3};\\ \text{mark}(2 \times ma \times mb + 16^2 + 1)\text{=MS4};\end{gathered}$$

Figure 3.10: State marker initialisation for Cb and Cr in L-CSPECK

$$\begin{gathered}\text{mark}(ma \times mb \times 1 : ma \times mb + 64) = \text{MIP};\\ \text{mark}(ma \times mb \times 8^2 + 1)\text{=MS3};\\ \text{mark}(ma \times mb \times 16^2 + 1)\text{=MS4};\end{gathered}$$

Figure 3.11: State marker initialization for both Cb and Cr in proposed algorithm

purposes. The meaning of each marker is as follows:

- MIP : the pixel is insignificant and will be tested for this bit plane
- MNP : the pixel is newly significant, so it will not be refined for this bit plane
- MSP : the pixel is significant and will be refined in this bit plane
- MS^x : a block of size $2^x \times 2^x$ is to be skipped

MS^x can also be defined as an edge marker to keep track of which pixels are insignificant. The value of MS^x in this algorithm is from MS1 to MS10. Based on Figure 3.8, MIP is marked for the lowest subband in the pyramids of coefficients. MS3 refers to the edge of the block with a size of $2^3 \times 2^3 = 64$. During

initialization, only markers from MS3 to MS10 (depending on image size) will be marked. This is crucial so that every edge is recognized for the partitioning process during the encoding and decoding process. Through the partitioning process, MS1 (block size 4) and MS2 (block size 16) will be updated automatically. A pre-computed constant lookup table, rather than calculation, is used during initialisation for faster implementation.

3.4.3 Sorting algorithm

The sorting algorithm for proposed algorithm is based on zeroblock partitioning similar to the original SPECK. Further explanation on sorting algorithm is in Section 3.4.4.

3.4.4 The proposed algorithm

The L-CSPECK and proposed algorithm are implemented using similar logarithmic structures. Since the location information is fixed, if the tested set is found to be significant, partitioning proceeds without the need to identify which plane belongs to the significant coefficients. The four stages of the algorithm are initialisation, the sorting pass, refinement pass and finally quantization step. During initialisation, scanning or linear indexing is first applied to all colour planes. This is the stage where the L-CSPECK and the proposed algorithm are applied differently. As previously explained in Section 3.4.1, scanning plays the key role in how the proposed algorithm is modified to enable the exploitation of similarities in the chrominance planes.

After scanning, marker for insignificant pixel, (MIP) is marked to all coefficients in the lowest subband of each plane. A state marker is first initialised based on the pre-computed constant lookup table. No list is required for this algorithm; and instead the state marker will be automatically updated if a significant coefficient is detected. Details of the pseudo code for the encoding process are given below. k in coefficient $C_{i,j,k}$ in the algorithm denotes the plane. The main difference between LCSPECK and the proposed algorithm occur during the initialization of the state marker and the mapping/scanning used.

Initialisation

1: Output $n = \lfloor log_2 (max |C_{i,j,k}|) \rfloor$

2: Map the transformed coefficient to match linear indexing
$seqt(map)$ = wavelet coefficients

3: Initialize the state marker
Mark NULL for all coefficients
$mark\,(all) = NULL$
MIP for the lowest subband coefficients (Figure 3.11)
$mark\left(1 : 2^{N-L}\right) = MIP$
MS ^{x} for each initial block or subband (Figure 3.11)
$mark\,(2^3 + 1) = MS3$

Sorting pass

The sorting pass is conducted through the test of significance in Equation (3.3). For chrominance planes, the sorting pass will automatically include both planes according to the interchanging scan order as shown in Figure 3.7.

1: For each coefficient with MIP mark, output $S_n\,(k)$;
if $S_n\,(k) = 1$, mark the coefficient as MNP, output sign of k.
For chrominance plane, if $n = odd, k = Cb$ and if $n = even$, $k = Cr$

2: For each coefficient with a non-NULL mark
if Set $S_n\,(S) \neq 0$
Partition the set into four equal subsets $O\,(S)$ (Figure 3.12).
Update the state marker by MS^{X-1}
or else
Skip the coefficient by marker size $MS^x = 2^x \times 2^x$

An example of partitioning and state marker updating can be seen in Figures 3.12 and 3.13. Figure 3.12 shows the initial marker set during initialization on both chrominance plane. Figure 3.13 shows the partitioning process where instead of two planes, the chrominance is treated as one plane. MIP is marked in both planes with interchanging order.

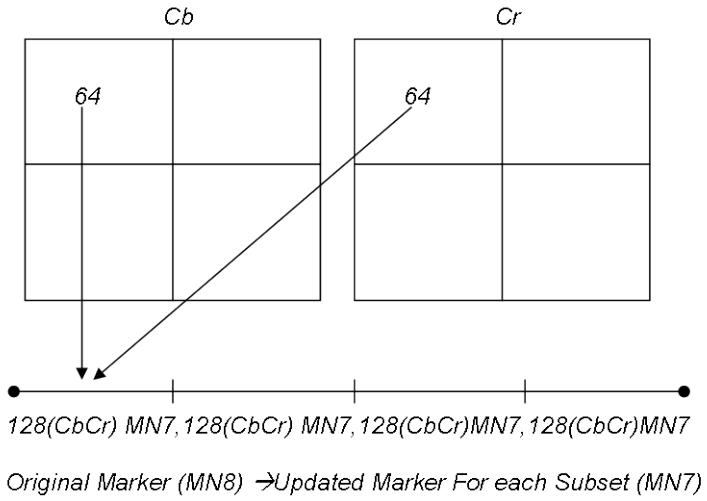


Figure 3.12: 4 Subset in chrominance plane

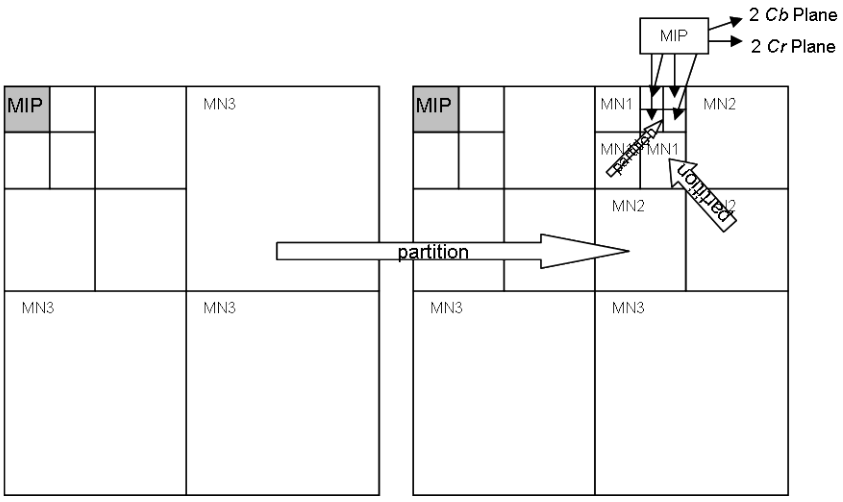


Figure 3.13: Partitioning and state marker update

Refinement pass

In the refinement pass, each coefficient marked as MNP will be refined to MSP and the marked MSP sends an output of the n^{th} most significant bit (MSB) of the coefficient to the bit-stream.

Quantization Step

Decrement n by 1 and continue to sorting pass $n = n_{max}, n_{max}-1, n_{max}-2, n_{max}-3, ...$

Significance tests are always done with bitwise AND, where the significance level for each bit plane is $T = 2^n$. The decoder will follow the same overall procedure as the encoder, but the test will use bitwise OR instead of bitwise AND. All operations can be implemented by bit shifting.

The proposed algorithm automatically tests both chrominance planes together because the generated scan order alternates between planes *Cb* and *Cr*. Suitable initial state markers are used by treating both chrominance planes as a single coded plane. This ensures that the partitioning process exploits the similarities between the *Cb* and *Cr* planes. Therefore, instead of duplicating the same structure of initial state markers to each plane, the proposed algorithm performs state marker initialisation once for both chrominance planes.

3.5 Numerical results and discussion

To assess the performance of the proposed coder, numerical experiments were carried out on a number of 24 bit colour images from an images database at USC-SIPI [62] and University of Granada [63]. The Barbara image [64] was cropped to size 512×512. For the purpose of discussion here, only a few images are presented with detailed results. This section describes the implementation of the proposed algorithm for lossy compression first and then for lossless compression.

For lossy compression, the RGB image was first transformed to a *YCbCr* colour space using a standard transform in Matlab (function *rgb2ycbcr*). Then, a CDF-9/7 tap biorthogonal wavelet filter with five decomposition levels [31] was applied to each colour plane. The coding performance of the proposed algorithm was evaluated using measurements of rate distortion, PSNR and structural similarity (SSIM). The distortion rate was measured in term of overall PSNR from the RGB plane after reconstruction, is represented by the Equations (3.4) and (3.5) where $C(i,j)$ and $\hat{C}(i,j)$ are the original and reconstructed images respectively at each colour plane and $N = N_x \times N_y$ is the number of pixels of the original image at

each colour plane.

$$PSNR = 10log\frac{3 \times 255^2}{MSE(R) + MSE(G) + MSE(B)} \tag{3.4}$$

$$MSE(C) = \frac{1}{N_x N_y}\sum_{i=1}^{N_x}\sum_{i=1}^{N_y}\Big(C(i,j) - \hat{C}(i,j)\Big)^2 \tag{3.5}$$

All decoded images for each bit rate were recovered from a single fidelity embedded encoded file truncated at the desired bit rate without any entropy coding. Back end arithmetic coding is expected to improve about 0.5 dB to the presented results [41, 42].

In lossless compression, the reconstructed image should be identical to the original image. The measurement of performance is based on the final bit-rate required to reconstruct the image. So, the compression ratio can be measured using:

$$Compression\ Ratio = \frac{bpp\ original\ image}{bpp\ compressed\ image} \tag{3.6}$$

The memory requirements during the coding and decoding processes are also presented to show the relative advantages in terms of memory disc-swapping saving using the proposed listless implementation.

3.5.1 Coding performance

The two types of measurement used for lossy compression are PSNR and SSIM measure.

PSNR measurement

Table 3.1 shows a comparative evaluation of the results obtained from CSPECK, the proposed algorithm and L-CSPECK. Also included are JPEG2000 and SPIHT, with the former was obtained from Kakadu v6.2 [38]. The analysis was performed at different rates, from 0.125 bits per pixel (bpp) for the compression ratio 192:1, to 3 bpp for the compression ratio 8:1. In theory, using a listless structure should slightly reduce the performance due to the use of fixed scanning, based on the

performance of the original listless implementation for a greyscale image [42]. However, the joint test of significance for chrominance planes in the proposed listless algorithm was expected to improve the listless performance to a level very close to that of the CSPECK.

The overall RGB PSNR was used for comparison, and therefore it can be seen that JPEG2000 show better results in most of the image. This was expected, since CSPECK partitioning shows a high PSNR only to luminance plane. Besides this, both JPEG2000 and SPIHT involve higher in computational complexity compared to to SPECK-based partitioning coder [37]. This means that the main comparison in Table 3.1 is between CSPECK, L-CSPECK and the proposed algorithm, with the best results given in bold. In most cases, the proposed algorithm outperforms the original CSPECK, which exceeds the expectations since the listless algorithm uses fixed memory and scanning. The difference is attributed to the bit savings from the joint chrominance test of significant.

It can be observed that the proposed coder improves the distortion rate over CSPECK at most rates on images that have good correlation between chrominance planes. The proposed algorithm maintains the zeroblock partitioning rule and jointly tests the chrominance planes to exploit the similarity between them. The value given below the name of each image is the correlation between planes Cb and Cr, measured using the function *corr*2 in Matlab. Note that if the value of this correlation value is low, the resulted PSNR shows better result in L-CSPECK or CSPECK because the joint test of the chrominance planes would give no advantage. The proposed coder also outperforms SPIHT at most rates, despite the fact that the latter uses more a complex zerotree partitioning rule and three lists. Surprisingly, compared with JPEG2000, the proposed coder performs better at certain rates in Lenna, Barbara and Baboon, but not with other images at most rates. This can be attributed to the simple joint chrominance test of significance generating more significant bits earlier in the bit-stream. Examples of the quality of reconstructed images from the proposed algorithm, L-CSPECK, CSPECK, SPIHT, and JPEG2000 are shown in Figures 3.14 and 3.15.

Table 3.1: Overall RGB PSNR performance for compressed colour image

Image	Bit Rate	CSPECK	Proposed	LCSPECK	SPIHT	JPEG2000
Lenna	0.125	27.3691	**27.6238**	27.4289	27.2939	27.9324
0.4522	0.25	29.6657	**30.0306**	29.6692	29.6245	30.0227
	0.5	31.5545	**32.3129**	31.6821	31.5999	31.7237
	1	33.0276	**34.1213**	33.1081	33.267	33.2909
	1.5	34.1498	**35.1304**	34.1699	34.3137	34.3181
	3	35.6755	**37.3529**	36.078	36.0136	36.0388
Barbara	0.125	24.8147	**24.8647**	24.813	24.473	25.2165
0.7303	0.25	**26.998**	26.9673	26.9386	26.8337	27.0782
	0.5	**30.0118**	29.9876	29.9591	29.8952	30.0859
	1	33.5733	**33.6082**	33.5844	33.5545	33.6865
	1.5	36.0107	**36.0622**	36.0215	35.9411	35.9189
	3	39.4147	**39.6262**	39.5932	39.7008	40.1194
Baboon	0.125	19.9299	**19.9836**	19.9764	19.8905	20.2485
0.7239	0.25	21.159	**21.1866**	21.1652	20.9723	21.3211
	0.5	22.4295	**22.486**	22.4773	22.5493	22.4234
	1	24.4759	**24.5034**	24.502	24.6004	24.3443
	1.5	25.7739	**25.8792**	25.8726	25.8621	25.5106
	3	28.4451	28.5779	**28.5883**	28.673	27.6762
House	0.125	23.5291	**23.5564**	23.5511	23.2223	24.2611
0.6507	0.25	25.6126	**25.7389**	25.7064	25.5832	26.4022
	0.5	28.0126	**28.1624**	28.1222	28.0987	28.4336
	1	31.1366	**31.2002**	31.1549	31.1053	31.0905
	1.5	32.8437	**33.16**	32.9974	33.0906	33.2032
	3	36.6188	**37.0766**	37.05	37.112	37.1758
Peppers	0.125	25.2197	25.2376	**25.2661**	25.0846	26.0629
0.2728	0.25	27.4656	27.4301	**27.5026**	27.654	28.4843
	0.5	30.0345	**30.2005**	30.1322	30.365	30.2517
	1	31.8663	31.9075	**31.9092**	32.3067	32.3334
	1.5	32.8575	32.987	**33.005**	32.9596	33.1652
	3	34.5775	34.8471	**34.8817**	34.9801	34.7091
Light House	0.125	30.4548	30.5576	**30.5584**	30.2664	31.3996
0.7248	0.25	32.6152	**32.7642**	32.7578	32.7378	33.4038
	0.5	34.7806	**34.8758**	34.8629	34.983	35.3198
	1	36.7404	**36.8024**	36.7914	36.9829	37.2593
	1.5	37.6434	**37.9972**	37.8722	37.9293	38.5563
	3	39.3987	**39.954**	39.9098	39.9579	40.8761
Butterfly	0.125	26.2682	26.3046	**26.3054**	25.9986	26.7484
0.0294	0.25	28.4053	28.4164	**28.4205**	28.2424	28.6146
	0.5	30.1962	**30.2713**	30.2691	30.3604	30.283
	1	32.0735	32.1438	**32.1482**	32.2632	32.2822
	1.5	33.1777	33.4069	**33.4279**	33.2204	33.2724
	3	35.4623	35.6413	**35.7478**	35.5948	35.0691
Ariel	0.125	22.2481	**22.25**	22.2481	22.1408	22.6824
0.6379	0.25	23.542	**23.5451**	23.542	23.5153	23.8517
	0.5	24.763	**24.8007**	24.763	24.7799	25.3024
	1	26.6029	**26.6096**	26.6029	26.7411	26.6165
	1.5	28.3549	**28.4523**	28.3549	28.2039	28.0783
	3	31.4894	**31.5206**	31.4894	31.4324	30.755

Structural similarity measurement

The measurement of structural similarity for colour images as mentioned in Section 2.2.4 is calculated after function 'rgb2gray' is applied to the reconstructed image [25]. The resulting SSIM values are shown in Table 3.2. Note that the bold value in the Table 3.2 is the maximum value between CSPECK, the proposed coder and L-CSPECK. As with the PSNR results, in most cases JPEG2000 performs best result except in few cases.
Although the improvements achieved in terms of SSIM and PSNR maybe considered small, the main advantage of the proposed system is the use of fixed working memory that enabling faster encoding and decoding. The introduction of combined test of significant for chrominance with listless technique also improves the results for certain image which exhibit good correlation between the chrominance planes. In cases where this correlation is low, CSPECK and L-CSPECK show better results than the proposed coder as with the previous results for PSNR [Table 3.1].

3.5.2 Memory requirement and algorithm complexity analysis

Similar to the original SPECK-based coder, the proposed algorithms first visits all pixels to gather information about bits in all bit planes (pre-process pass) [7]. This pass require one bitwise OR operation per pixel, following a predetermined sequence, and some analysis of partial results. Bit plane coding algorithms compute the same data to determine the first significant bit plane with at least one non-zero bit. Therefore, the complexity of bit plane coding can be measured approximately by counting the number of bits comparison (whether equal to 0 or 1) used to test the bits in the bit plane. CSPECK, L-CSPECK, and the proposed algorithm all use the most basic operations, such as memory access, bit shifting, addition for iteration, and bit comparison. L-CSPECK and the proposed algorithm also implement the same zeroblock partitioning rule, so that the level of complexity is the same as that of CSPECK. By jointly testing the chrominance planes, the proposed coder actually lowers the complexity associated with bit comparison in the algorithm. This is due to the fact that it uses fewer signifi-

Table 3.2: SSIM measure for compressed colour image

Image	Bit Rate	CSPECK	Proposed	LCSPECK	SPIHT	JPEG2000
Lenna	0.125	0.8742	**0.8905**	0.8841	0.8663	0.8981
0.4522	0.25	0.9254	**0.9394**	0.9315	0.9234	0.9446
	0.5	0.9551	**0.9628**	0.9561	0.9551	0.9689
	1	0.9764	**0.9787**	0.9754	0.9734	0.984
	1.5	0.9813	**0.9897**	0.9867	0.9813	0.9899
	3	0.991	**0.9941**	0.9903	0.9895	0.9962
Barbara	0.125	**0.8144**	0.8142	0.8133	0.8025	0.8428
0.7303	0.25	0.8974	**0.9005**	0.9001	0.8807	0.9063
	0.5	0.951	**0.9548**	0.9546	0.9437	0.9572
	1	0.9826	**0.9829**	0.9828	0.9781	0.986
	1.5	0.9889	**0.9922**	0.9919	0.9885	0.9925
	3	0.9966	**0.9977**	0.9976	0.9965	0.9982
Baboon	0.125	0.6267	**0.6522**	0.6508	0.5873	0.6511
0.7239	0.25	0.7335	**0.737**	0.7338	0.7276	0.7998
	0.5	0.8558	**0.8701**	0.8698	0.8419	0.8884
	1	0.9187	**0.9418**	0.9418	0.919	0.9492
	1.5	**0.9563**	0.955	0.9549	0.9465	0.97
	3	**0.9856**	0.9849	0.9849	0.9785	0.9903
House	0.125	0.8309	**0.8325**	0.8325	0.805	0.8431
0.6507	0.25	0.8945	**0.9055**	0.9041	0.8782	0.9165
	0.5	0.9352	**0.9512**	0.9492	0.9327	0.957
	1	0.968	**0.9737**	0.973	0.967	0.9801
	1.5	**0.9846**	0.9839	0.9839	0.9799	0.9907
	3	**0.9941**	0.9938	0.9938	0.9924	0.997
Peppers	0.125	**0.8678**	0.8673	0.8673	0.7967	0.8894
0.2728	0.25	**0.9234**	0.9229	0.9228	0.8686	0.9405
	0.5	**0.9552**	0.9545	0.9546	0.9152	0.9688
	1	0.9752	**0.9758**	0.9758	0.946	0.9849
	1.5	**0.9769**	0.9759	0.9759	0.9515	0.99
	3	**0.9911**	0.9905	0.9904	0.9715	0.9962
Light House	0.125	0.9073	0.9102	**0.9103**	0.8913	0.9193
0.7248	0.25	0.9485	**0.9528**	0.9528	0.9428	0.9593
	0.5	0.971	**0.9748**	0.9747	0.9705	0.9791
	1	0.9834	**0.9881**	0.9881	0.9844	0.9898
	1.5	0.9908	**0.9901**	0.9901	0.9888	0.9942
	3	0.9956	**0.9961**	0.9961	0.9949	0.9981
Butterfly	0.125	0.8599	**0.8681**	0.8681	0.8476	0.8843
0.0294	0.25	0.9221	0.9234	**0.9235**	0.9191	0.9428
	0.5	**0.9606**	0.9601	0.96	0.9585	0.971
	1	0.9794	0.981	**0.981**	0.9784	0.9866
	1.5	0.9822	0.9833	**0.9838**	0.9822	0.9913
	3	**0.9935**	0.9932	0.9932	0.9926	0.9973
Ariel	0.125	0.7591	**0.7635**	0.7634	0.7558	0.7752
0.6379	0.25	0.8065	**0.8155**	0.8153	0.8118	0.8231
	0.5	0.8405	**0.8493**	0.8486	0.8624	0.8736
	1	0.8897	**0.8909**	0.8908	0.9063	0.9252
	1.5	**0.9346**	0.9343	0.9337	0.9335	0.9404
	3	0.9628	**0.972**	0.9709	0.9706	0.9731

(a) Original (b) JPEG2000 27.0782 dB (c) CSPECK 26.9980 dB

(d) SPIHT 26.8337 dB (e) L-CSPECK 26.9386 dB (f) Proposed 26.9673 dB

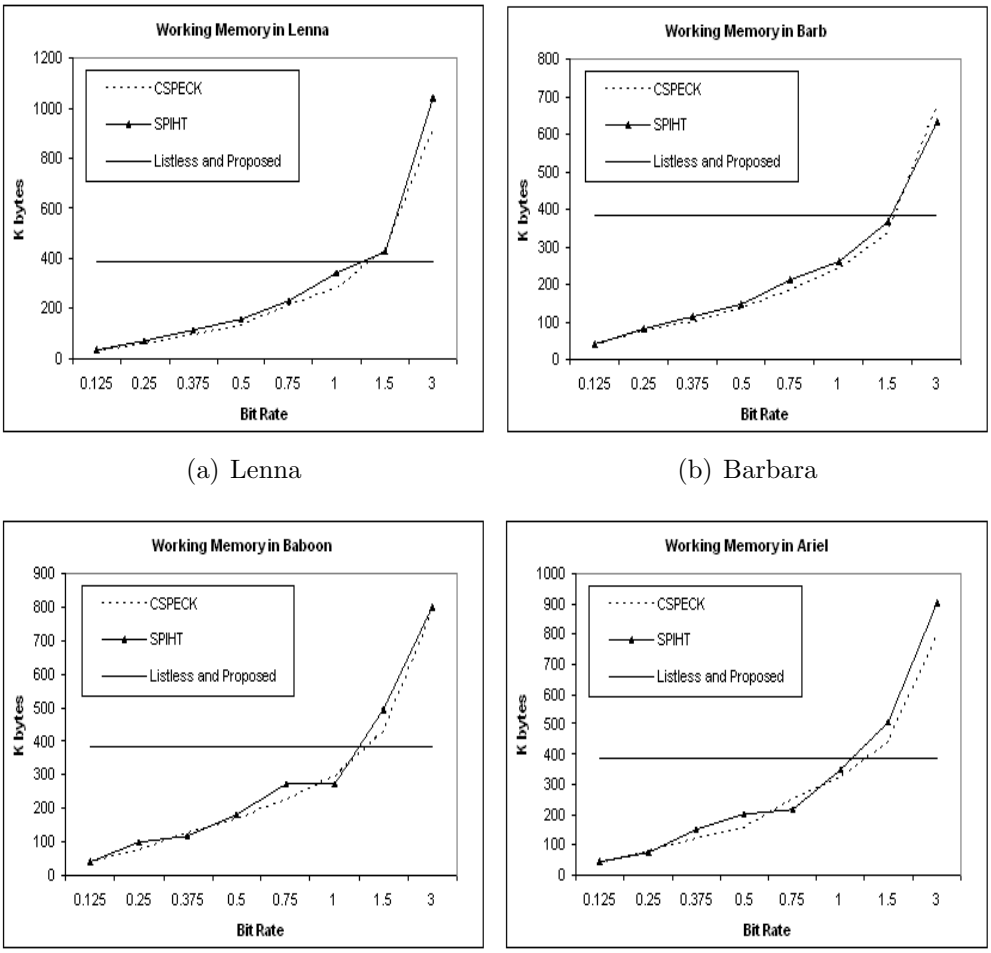
Figure 3.14: Reconstructed Barbara at rate of 0.25 bpp

(a) Original (b) JPEG2000 33.2909 dB (c) CSPECK 33.0276 dB

(d) SPIHT 33.267 dB (e) L-CSPECK 33.1081 dB (f) Proposed 34.1213 dB

Figure 3.15: Reconstructed part of Lenna at rate of 1 bpp

cance tests if the jointly tested planes give insignificant results. CSPECK has a lower level of complexity compared to SPIHT and JPEG2000, and so the proposed algorithm has the lowest complexity of all due to its joint significance test in the chrominance plane. The algorithm efficiency and speed of operation can



(a) Lenna (b) Barbara

(c) Baboon (d) Ariel

Figure 3.16: Working memory comparison

also be attributed to the memory needed during the coding process (see Figure 3.16). Memory requirements in CSPECK also depends on lists that may cause data dependant, variable memory requirements and memory management as list nodes are added, removed and moved. The proposed algorithm avoids any such memory management, because the memory used is fixed, there are no lists and a

marker is only updated each time significant coefficient is found. The marker requires 4 bits, so the memory needed for the listless algorithm (either L-CSPECK or the proposed coder) can be calculated based on equation below:‘

$$M_{listless} = Size\,of\,Image\,(x \times y) \times 3 colour\,planes \times 4\,bits \tag{3.7}$$

As the memory is fixed, the amount of memory at lower bit rates is more in listless than the coder with list. However, the efficiency of listless algorithm is better because its fixed memory allocation. Furthermore, as bit rates becomes higher, it become more efficient than SPIHT and CSPECK due to the growing size of list with those algorithms. The memory requirements for CSPECK and SPIHT are calculated using Equations 3.8 and 3.9 for each rate:

$$\begin{aligned} M_{SPIHT} &= (L_{LSP} + 2 \times L_{LIS} + L_{LIP}) \\ &\quad \times (bitX + bitY + bitZ) \end{aligned} \tag{3.8}$$

$$\begin{aligned} M_{CSPECK} &= (L_{LSP} + 2 \times L_{LISY} + L_{LISI} + L_{LISQ}) \\ &\quad \times (bitX + bitY + bitZ) \end{aligned} \tag{3.9}$$

where; L refers to the number of coefficients in the list, $bitX$ and $bitY$ refer to the number of bits required to represent the coordinate for each plane, and $bitZ$ refers to the colour plane.

Based on graphs in Figure 3.16, it can be seen that the memory increases with the increase of the bit rate. The memory of the proposed algorithm is fixed and represented by a straight line in the graph. The dynamic memory of SPECK and SPIHT is directly proportional with an increase of the bit rate. At the same time, dynamic memory means that the coding needs memory management in order for the list required to be added and removed during coding process. On the other hand, the fixed memory allows the encoding and decoding processes to operate faster because memory management is removed from coding process. Since the memory is fixed for any rate, the lossless application will be greatly beneficial when applied with this method. The next section describes lossless

implementation using this structure.

3.5.3 Lossless performance

To implement the lossless coder, the reversible colour transform (RCT) is used. Then a reversible wavelet 5/3 tap filter is applied to each plane. The performance results are given in Table 3.3 and, based on the compression ratio achieved, JPEG2000 gives the best results for all images. However, in general, the proposed algorithm outperforms both SPIHT and CSPECK. It provides outstanding savings on the memory required for the encoding and decoding processes and these saving will greatly reduce the time required to perform the compression. Memory is calculated based on the final list generated using the Equations 3.7- 3.9. Table

Table 3.3: Lossless Performance Based on Compression Ratio

Colour Image (24 bits)	JPEG 2000	Proposed	L-CSPECK	CSPECK	SPIHT
		512x512			
Lena	1.7012	1.6878	1.6355	1.6356	1.5980
Barbara	1.8078	1.7214	1.7212	1.7249	1.6994
Peppers	1.6215	1.5420	1.5512	1.5513	1.5126
Ariel	1.4548	1.4223	1.4207	1.4208	1.4039
Baboon	1.3273	1.3033	1.3075	1.3075	1.2859
		256x256			
Barche	1.9773	1.8711	1.8665	1.8674	1.8475
Peppers	1.9435	1.8270	1.8269	1.8274	1.8234
Couple	1.9714	1.8926	1.8993	1.9003	1.8679
Tulips	1.6116	1.5534	1.5583	1.5583	1.5576
Jelly	2.4947	2.2869	2.2727	2.2737	2.2638
Average	**1.7911**	**1.7108**	**1.7060**	**1.7067**	**1.6860**

3.4 shows that the proposed listless coder offers savings of more than 70 % of the working memory required for both SPIHT and CSPECK. No comparison is made to JPEG2000 because the algorithm uses difference technique that include layered block coding, fractional bit planes and block based rate distortion optimization. Such huge saving will greatly benefit hardware implementations of progressive lossless compression. Although there are better techniques for loss-

Table 3.4: Memory requirement for lossless(in bits) & saving

Image	CSPECK	SPIHT	Proposed	% saving CSPECK	% saving SPIHT
		512x512			
Lena	15658440	15682200	3145728	79.9103	79.9407
Barbara	15667620	15719560	3145728	79.9221	79.9884
Peppers	15698580	15707920	3145728	79.9617	79.9736
Ariel	15728460	15728520	3145728	79.9998	79.9998
Baboon	15728460	15728560	3145728	79.9998	79.9999
		256x256			
Barche	3526038	3531060	786432	77.6964	77.7281
Peppers	3531816	3534336	786432	77.7329	77.7488
Couple	3465990	3483792	786432	77.3100	77.4259
Tulips	3536892	3537684	786432	77.7648	77.7698
Jelly	3463668	3497256	786432	77.2948	77.5129

less compression that are able to produce much higher compression ratio, the progressive to lossless coding can benefit to applications where the operator can views the approximate image before the final lossless image transmitted.

3.6 Conclusions

In this chapter, an efficient and simple colour image coding algorithm has been developed. The algorithm has been tested for both lossy and lossless compression. The SPECK-based listless algorithm known as LsK [42] was expanded for composite colour image compression, known as L-CSPECK. Although colour images can be compressed separately, composite techniques provide precise bit control and exploitation to nonlinear dependencies at high transition regions (such as edges) that remain among the spectral planes. This proposed algorithm has interesting properties which offer reduced working memory, but maintains level of performance and efficiency similar or better than CSPECK. The L-CSPECK was then modified to exploit the similarity in chrominance planes in order to offer better performance in terms of rate distortion and final bit rates for images that have good correlations between the two planes. This was achieved through a simple modification of the scanning order so as to combine test of significance for the chrominance planes. All tested algorithm is bit planes coded to gener-

ate a fidelity embedded bit-stream for colour images. The results for JPEG2000 were superior to those of the other algorithms tested, which was as expected, because it uses block based wavelet compression which involves higher computational complexity. Although the implementation of the listless structure reduced the working memory required during the coding process, the proposed algorithm managed to outperform the SPIHT on most tested images at low bit rates for lossy compression. A detailed comparison with the CSPECK and L-CSPECK using measures of PSNR and SSIM showed that the proposed algorithm outperforms the other methods with images that exhibited high correlations between chrominance planes at certain bit rates. When the tested joint chrominance plane are insignificant in the proposed coding more bits are saved for other significant coefficients. For progressive lossless compression, the fixed memory used in both listless colour image coders (L-CSPECK and the proposed algorithm) is very low compared to the final dynamic memory required using CSPECK and SPIHT about 70% saving. The huge savings found for the proposed algorithm can greatly benefit its performance, and this is especially significant for hardware implementation. In conclusion, the proposed listless structure for colour images offers tremendous working memory savings compared to conventional rate scalable coders while maintaining similar or better level of performance compared to state of the art algorithms such as SPIHT and CSPECK .

Chapter 4

Listless Implementation for Embedded 3D Image Coder

4.1 Introduction

Three dimensional (3D) data is now common in many fields. Advances in the technology of data acquisition in medicine and geography for example, have led to an exponential growth in the availability of 3D image data. Other factors that contributing to these developments include storage technology and interactive computer applications that use 3D models in response to the demand for realism in graphic representation. The graphic details are represented by a huge amount of data. Compression is essential to allow the efficient transfer and storage of these data. The general idea of transform-based compression is to use a transform that decorrelates the input signal and packs its total energy into a small number of coefficients. Two methods of transform-based compression that have good decorrelation properties are dicrete cosine transforms (DCT) and discrete wavelet transform (DWT). 3D compression can be considered to be quite similar to video compression where the third dimensions is time. This chapter discuss three applications that uses three dimensional transform for compression. The applications are video, medical and remote sensing imaging (multispectral images).

Motivation

This chapter proposes an efficient algorithm for 3D image compression based on 3D-SPECK. 3D images source such as volumetric medical images can be compressed more efficiently in the 3D tranform domain. Although compression in two dimensional domain can be applied to a 3D image source, the implementation in the 3D domain will benefit from further compression in the third dimension, such as the spectral domain or the time domain for frames in video. This will increase the compression performance of a 3D source.

During the compression process, the quantisation of data that has been decorrelated using suitable transforms is crucial. In the wavelet domain, bit plane coding is the most attractive scalar quantisation method, because it is very simple but still works efficiently to reduce the size of the image representation. The implementation of 3D wavelet transforms with 3D images source highlights the need for suitable coding techniques during quantisation. One available type of technique extends the appropriate 2D algorithm, for example, the 3D SPECK and 3D SPIHT. The problem with these implementations is the introduction of lists which require dynamic memory for location information to be added, removed and moved during the coding process. As a consequences, the algorithm depends upon the management of variables, data dependency and memory. As the coding process proceeds, the two lists in 3D SPECK and the three lists 3D SPIHT grow in size, and this can lead to considerable inefficiency in the compression system. This present study is motivated by efficient listless algorithm introduced for colour images (Chapter 3). Hence, the idea here is to use the fixed working memory approach during the coding process. This would enable faster coding and decoding with reduced memory requirements compared to the original 3D SPECK coding. The proposed 3D algorithm operates without a linked list and is suitable for implementation in hardware applications. The 3D listless SPECK has a fixed, predetermined memory requirement that is larger than that required for the image alone. In the new algorithm developed here, rather than lists, a state table with 4 bits per coefficient keeps track of the set partitions and the information that has been encoded. Sparse marking is applied to the selected block nodes of insignificant blocks in the state table. This way, a large group of predictable

insignificant coefficients can be identified and skipped during the coding process. The performance of the proposed listless algorithm will then be compared with that of original 3D SPECK.

4.2 3D SPECK coder

The 3D SPECK was proposed by Tang and Pearlman [65, 66]. The extension to 3D implementation should be suitable for 3D image sources and enable the exploitation of correlations in third dimensional domains, for example, the time domain in video and the spectral domain in multispectral remote sensing images. Wavelets have been shown to be very adaptable with a wide range of data with reasonable level of complexity. The 3D discrete wavelet transform (DWT) transform domain can exploit correlations in the multispectral images which exhibit tight statistical dependency along the wavelength axis of the prism [67].The 3D SPECK maintains a block splitting algorithm to sort the significant pixels. If a code block contains significant coefficients it is split into smaller sub-blocks. This technique can zoom into areas with high energy and code these first, which is a suitable method for exploiting the presence of significant high frequency intra-band components.

The 3D SPECK is expected to be an excellent candidate for use with multispectral images due to their properties of concentrating energy in the high frequency band. The rate distortion curves have proved that SPECK algorithms out-perform the SPIHT for images with higher frequency content. This was proved to be true in both 2D and 3D cases in the initial SPECK development [37, 65, 7, 68]. The 3D SPECK maintains two link lists, the LIS and LSP [66]. It comprises of three main stages, initialization, sorting and refinement. The sorting pass is based on the block splitting method after a test of significance. In the refinement pass some of the coefficients are transmitted, and the quantisation process continues for the next bit plane (decreased) until certain bit rate achieves. The structure of the 3D SPECK is shown in Figure 4.1[65]. The finest pixels lie at the upper level of the pyramid while the coarsest pixels lie at the lowest level (root).

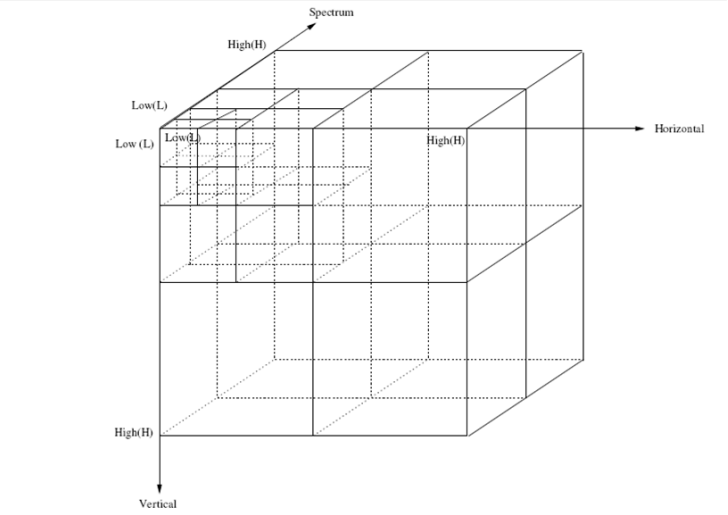


Figure 4.1: Structure of 3D SPECK

4.3 3D listless SPECK

The idea for the 3D Listless SPECK came from the implementation of the listless SPECK algorithm for colour images. The transformed coefficients in the 3D wavelet domain decorrelate the multispectral image components spatially and spectrally. These coefficients display some redundancy that can be exploited during the coding process. The proposed coder in this study follows the logic of a basic sorting algorithm, as in the 3D SPECK. So that, inter-band dependence can be exploited automatically. The first modification is that the list is removed, and to compensate for that, a state table marker is used to identify and keep track of the coefficients. Another modification relates to the indexing of the one dimensional array, which offers some computational and organisational advantages. This technique can be considered to be similar to the algorithm applied to 2D images [41, 42, 44, 60, 57, 69].

The linear indexing system uses a single number, rather than three to represent the index of coefficient. To map the 3D coefficients onto a one dimensional array, the mapping is based on a recursive Z curve or Morton ordering. This indexing is used instead of a raster scan because it performs better in preserving the locality of the data points. This type of ordering also suits the pyramidal structure of wavelet coefficients. The linear indexing or also known as Z-curve/Morton order

for three dimensional structure is shown in Figure 4.2[70]. Linear indexing efficiently supports the operations carried out on coefficient positions needed for tree based algorithms with one operation instead of three, assuming the usual subband arrangement of 3D wavelet coefficients. Following the linear indexing, the coeffi-

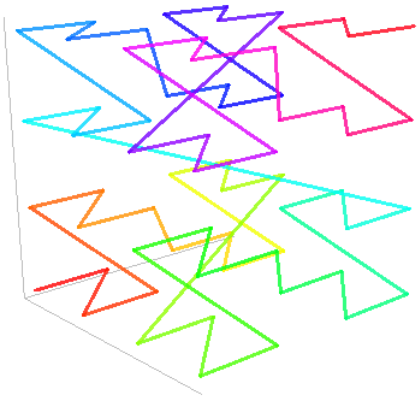


Figure 4.2: Morton scan in 3D

cients is stored in a single array of length I, where $I = Row \times Column \times Frame$. This is the magnitude array. The state table array, which is based on linear indexing, also has length I, with the marker uses 4 bits per coefficient. There is a one-to-one correspondence between magnitude and mark. The memory needed for this proposed algorithm is fixed, depending on the size of the image. For example, a total of 1 MB of memory would be required to use this algorithm for 8 bpp image, size of $128 \times 128 \times 128$ with an optional pre-computed array. This actually gives a total savings of 67% compared to the memory needed for the original 3D SPECK algorithm. The original 3D SPECK uses dynamic memory that based on the size of list generated for the coding. With the use of two lists, the total dynamic memory requirement might amount to 9 MB to store the location information in the list. State table markers are placed in a 4 bit coefficient for each marker, to keep track of the set partitions. Each element of the mark signifies a cube, indicating something about the corresponding element in the values in the image array. Each marker and its meanings are shown below:

- MIP: the pixel is significant

- MNP: the pixel is newly significant, so it will not be refined for this bitplane
- MSP: the pixel is significant and will be refined in this bitplane
- $MS2$: a block of size $2 \times 2 \times 2$ (i.e., 8 elements) is to be skipped
- $MS4$: a block of size $4 \times 4 \times 4$ (i.e., 64 elements) is to be skipped

An additional pointer is added to reduce the computational time required to calculate the number of MS2 markers. In the sorting pass, the algorithm performs a significance test on the MIP, followed by test for all of the other blocks. All of the first elements of the initial block are marked by a marker and then it is partitioned by checking its significance. A block is significant if at least one of its coefficients is equal to or greater than the threshold. These are then partitioned into smaller sets for further significance testing. The notation below is based on the original 3D SPECK algorithm in order to describe the structure and terminology of the new proposed algorithm. S is significant with respect to n, if :

$$max_{i,j,k \epsilon S} |c_{i,j,k}| \geq 2^n \tag{4.1}$$

where $c_{i,j,k}$ denotes the transformed coefficients at coordinate (i, j, k). Otherwise it is insignificant. The significance can be defined as follows:

$$S_n(T) = \begin{cases} 1 & if max_{i,j,k \epsilon T} |C_{ijk}| \geq 2^{n+1} \\ 0 & otherwise \end{cases} \tag{4.2}$$

As with the 3D SPECK, the coding process consists of three main stages, initialisation, the sorting pass and refinement. Now we can define the encoding algorithm for the 3D listless SPECK:
Initialisation:

1. Output $n = \lfloor log_2 (max|c_{i,j,k}|) \rfloor$
2. Map the transformed coefficient to linear indexing:
   $S_n(map3d) = \lfloor c_{i,j,k} \rfloor$
3. State table initialisation:
   - MIP for LLL (low-low-low) subband (as can be seen in Figure 4.1)

- MSX for each initial tree; where the marker defines numbers of coefficient to be skipped. For example, MS2 is equal to $2 \times 2 \times 2 = 8$ coefficients skipped. (X refers to a cube of size 2,4,8,16)
- The state marker is initialised for listless implementation. This marker is set fixed based on assumption that it match the generated wavelet coefficients following the Z-curve/Morton order.

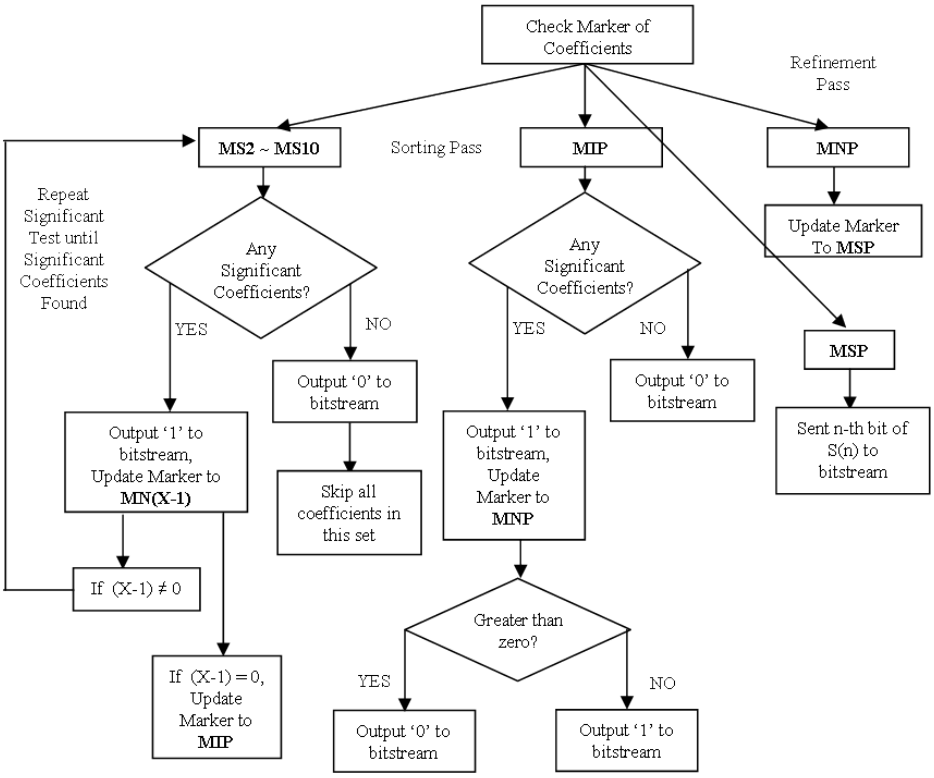


Figure 4.3: Sorting pass for listless 3D SPECK

The sorting pass:
The flowchart for the sorting pass stage is shown in Figure 4.3. In increasing order of linear indexing, the sorting pass algorithms are as follows:
Significance test based on Equation 4.2, for each coefficients:
If coefficients are marked MIP $S_n\,(MIP)$:

1. Output $S_n\,(MIP)$, whether or not the set is significant with respect to the current threshold,2^n

2. If $S_n\,(MIP) =' 1'$, output sign of S_n and update marker to MNP, $S_n\,(MNP)$. Otherwise, test next coefficients.

If coefficients are marked MSX where X is between 2 and 10, $S_n\,(MSX)$:

1. Run significance test on the marked block.
   For example $S_n\,(MS3)$, so the number of coefficients to be tested as a block is $2^3 \times 2^3 \times 2^3 = 512$.
2. Output $S_n\,(MSX)$, whether or not set coefficients are significant respective to current threshold 2^n
3. If $S_n\,(MSX) =' 1'$, partition the current set into 8 subsets with size $2^{n-1} \times 2^{n-1} \times 2^{n-1}$ and update each initial set marker to $MS\,(X-1)$, $S_n\,(MS\,(X-1))$, or else skip the whole set of coefficients.

Refinement pass:

1. For each entry marked as $S_n\,(MSP)$, except for those included in the last sorting pass, output the n^{th} MSB of $S_n\,(MSP)$.
2. If entry is marked as $S_n\,(MNP)$, update the marker as $S_n\,(MSP)$.

Quantisation step:
Decrease n by 1, $n-1$ and perform sorting pass until the required bit rates achieves. Significance tests are conducted using the bitwise AND operation. All operations can be implemented by bit shifting.
Further entropy coding, based on the statistical properties of the generated bit-stream, will improve compression performance up to 0.5 dB [41, 42]. However, for simplicity, the test is carried out without any entropy coding applied to the generated bit-streams. The decoding process will follow the same logarithmic order to the input bit-streams, and the image can be reconstructed progressively when the required bits rate are achieved. This process can also be used for lossless compression when used with integer wavelets during the transformation process.

Algorithm complexity

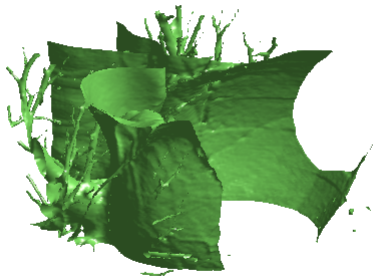
The partitioning process used in the proposed algorithm follows a similar concept to that of 3D SPECK partitioning rule, based on zeroblock partitioning. The

algorithm maintains the lowest level of complexity compared to other types of 3D-based embedded coding such as the 3D SPIHT or the JPEG2000 multicomponents. For example, the 3D SPIHT algorithm accesses the vales of coefficients with particular zerotree order which is quite complex, in order to search for significant coefficients among the entire transformed coefficients. This unstructured coefficients access in 3D SPIHT algorithm hinders its use in certain memory constrained environments. Subsequently, as with the previous implementation of the listless algorithm to colour image discussed in Chapter 3, the key contribution in the implementation of a listless algorithm with 3D SPECK is the reduced working memory required when the need for a dynamic list is removed from the encoding process. To describe the efficiency, we will compare the amount of required memory when listless coding are applied. The dynamic memory in the original 3D SPECK is required so that the list can store the coordinates that represent the positions of the sets of coefficients tested and of the significant coefficients found. This dynamic memory is small at lower bit rates, but as bit rates increase or in application where images must be reconstructed with much higher precision, as in medicine and the remote sensing of multispectral images, the memory needed becomes very large, subsequently leading to slower encoding and decoding processes.

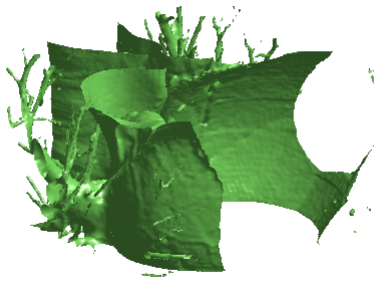
As previously mentioned in Section 4.2, the original 3D SPECK is expected to work very well on images with high frequency content, such as multispectral remote sensing images. This assumption is based on the performance of the original 2D SPECK with the same type of images such as Barbara, compared to zerotree-based algorithms like the SPIHT. However, due to its efficient implementation, the 3D listless SPECK can be applied to various image type, for example video source or volumetric medical images. On the other hand, in terms of rate distortion or final bit rate, its performance might be not as good as other embedded coder such as the Joint Photographic Experts Group (JPEG)2000 multicomponent and the 3D SPIHT. Therefore, the comparison of performance will be limited to the listless algorithm and the 3D SPECK only.

4.4 Numerical results

The performance of the proposed coder, 3D listless SPECK is evaluated based on three measures: firstly, the working memory used during coding or savings in terms of required memory space; secondly, its rate distortion PSNR measured for lossy compression; and finally, the bit rate or compression ratio for lossless compression. The 3D images are taken from various sources, such as, video [71, 72], volumetric medical images [73, 74] and remote sensing satellite images [75, 76]. For an example, Figure 4.4 shows the reconstructed Vessel image at 0.5 bit rates compressed using 3D SPECK (viewed using Matlab function 'isosurface').



(a) Original Vessel



(b) Compressed at 0.5 bit rate

Figure 4.4: Compression of volumetric Vessel image

4.4.1 Working memory

The proposed algorithm operates without a linked list, and so instead has a predetermined/fixed memory requirement. A state table with 4 bits per coefficient keeps track of the partitioning process and the encoded information. For 3D image with size of 8 bits, using the proposed algorithm means that the memory requirements are fixed at approximately 50% of the 3D image's original size, and this can be calculated using the equation below:

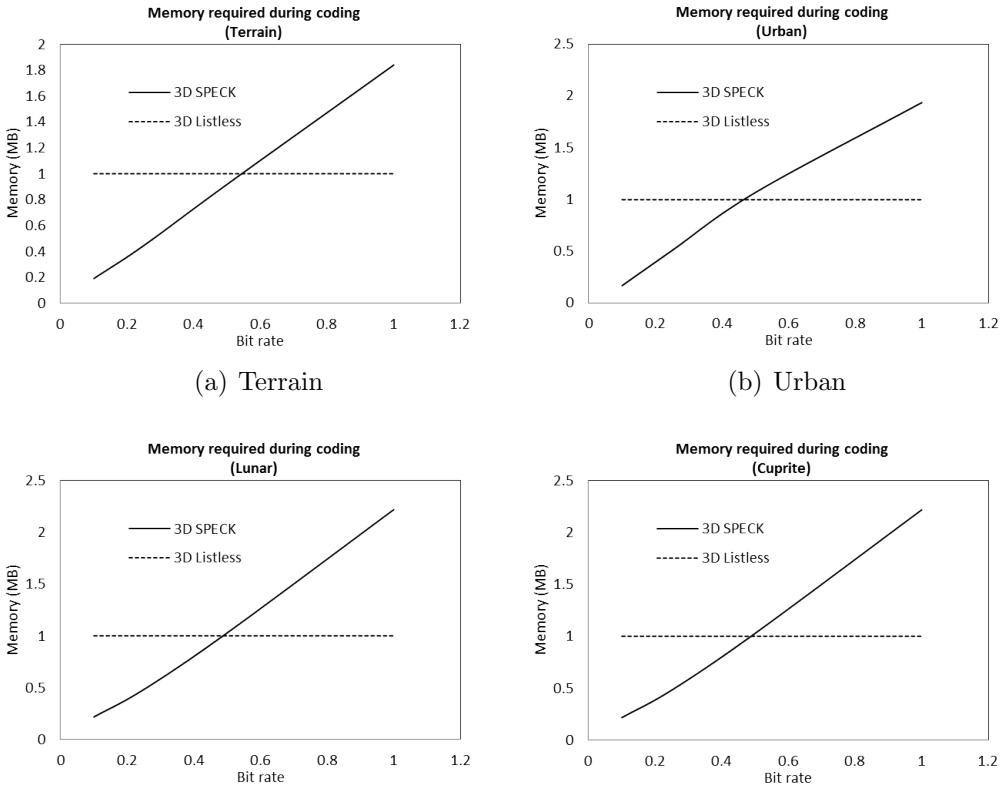
$$\begin{aligned} \text{image dimension} &= height \times width \times \text{number of frames/slice} \\ Memory_{listless} &= \text{image dimension} \times 4bits \end{aligned} \tag{4.3}$$

For the 3D SPECK, the memory required is based on the number of lists and the bits required to represent the coordinate of coefficients in the list. So, similar to Equation 3.9, the memory required to code the 3D SPECK is:

$$Memory_{3dSPECK} = (L_{LSP} + L_{LIS}) \times (bitX + bitY + bitZ) \tag{4.4}$$

where
L refers to the number of coefficients in the list,
$bitX$, $bitY$ and $bitZ$ refer to the number of bits required to represent the coordinates for each dimension
So, for example for an image of size $128 \times 128 \times 128$, the bits required to represent each plane are $log2(128) = 7$ which gives a total of 21 bits for each item in the list. The final lists generated at a bit rate of 0.5 are $L_{lsp} = 166552$ and $L_{lis} = 354109$, so the total memory in bits is equal to 10933881 bits or 1.3034 MBytes. For the listless implementation, the memory required for any compressed image in lossless compression is $128 \times 128 \times 128 \times 4 = 8388608\, bits$ or 1 MB. Although this still seems large, its fixed nature means that the algorithm requires no memory management where list updating the lists requires adding or removing of items. Moreover, the list in the SPECK is calculated for the lower rate of 0.5, which means that the memory required will increase at higher rates. For example, at a bit rate of 1, the memory needed for lists generated in the 3D SPECK is 2.3313 MB. Yet the listless implementation requiring a fixed

memory is applicable to all rates. At 0.5 and 1 bit rates, the memory savings of Vessel image are 23.2788% and 57.1055% respectively compared to the memory used in the original 3D SPECK implementation. These saving will become larger and give advantages greater especially in applications which require lossless and progressive coding, as with the implementation for colour images in Chapter 3. Graphs in Figure 4.5 show the increasing memory required in 3D SPECK when compared with the fixed listless algorithm.



(a) Terrain (b) Urban

(c) Lunar (d) Cuprite

Figure 4.5: Comparison of memory required during coding process

4.4.2 Comparison of lossy compression performance

During initial tests, the overall performance of the 3D listless coder was tested on randomly selected 3D sources for comparison with the 3D SPECK. Here, a popular video sources for compression are cropped and volumetric medical

images are tested for lossy compression. The 3D transform used is CDF-9/7 filter with five level decompositions applied to each dimensions. Lower levels of decomposition are applied to certain images mentioned in the Table 4.1, where compression results are shown. The rate distortion, PSNR performance of the reconstructed 3D image are measured using the equation below [65]:

$$PSNR = 10log_{10}\frac{\left(2^{nsb}-1\right)^2}{MSE} \tag{4.5}$$

where nsb denotes the maximum bit depth, and MSE is the mean square error between the original and reconstructed image.

From the results shown in Table 4.1, the PSNR of the proposed 3D algorithm perform better at certain rates, especially with images with five levels of decomposition in each dimensions. Using different level of decomposition on images such as Saggital and Coronal generate a slight decay in PSNR. The decay rate close to 1 dB, where in both images three level of decomposition are applied. This decay are expected since the proposed algorithm uses fixed initial state markers on all images. These marker are assumed to matched with the indexing that mapped similar to five level of decompositions at each dimension. An adaptive state marker initialization can reduce the decaying results, but come at the cost of additional calculation/complexity to coder. Images applied with five level decompositions produce small improvement at certain rates around 0.1 dB are evident, which suggests a performance similar to that of the 3D SPECK for natural 3D images type (video and medical source).

Multispectral images

Hyperspectal data commonly possess very good spatial resolution with superfine spectral details, but these come at the cost of large datasets. For example, with 224 subbands formed with 16 bit representation, the amount of data for an image size of 512×614 pixels is about 134MB. Therefore, efficient compression needs to be applied for transmission and storage. In this study, only few selection of spectral band uses for compression, so the suitable term uses is multispectral images, hyperspectral refers to the original remote sensing image, where source

Table 4.1: Rate distortion performance with various types of 3D source

Bit Rate	3D SPECK	Proposed 3D Listless
	Miss Am	
0.1	35.3025	35.3865
0.125	36.0702	36.0948
0.25	39.4489	39.4419
0.5	43.4085	43.5750
	Mri	128x128x32
0.1	20.2952	20.4054
0.125	20.7778	20.9662
0.25	23.1532	23.1906
0.5	26.3266	26.1727
1	30.7402	30.9200
	Vessel	128x128x128
0.1	29.9591	29.9833
0.125	30.5863	30.7609
0.25	33.1563	33.2475
0.5	35.1249	36.1777
1	39.3679	39.3904
	Saggital 3 level	256x256x16
0.125	36.6471	36.6471
0.25	44.6971	43.5060
0.5	56.8300	55.4794
	Coronal 3 level	256x256x16
0.125	34.4903	34.0668
0.25	41.2838	40.1890
0.5	51.7288	50.1832

taken. Figure 4.6[75] represent the the spectral properties in hyperspectral images, whereas multispectral consist of fewer spectral bands than the hyperspectral images. Multispectral images have specific properties that can be used in an efficient compression system. Another example of a hyperspectral image source is Terrain [76] with an original size of 307 pixels by 500 lines by 210 band which is usually represented using 16-bit precision, requiring 61.48 MB for storage. Compression of this type of 3D image sources usually requires the exploitation of interband correlation. Hyperspectral/multispectral images usually contain high frequency information, which means that the 3D SPECK is likely to be an excellent contender which will offer a lower level of complexity than other progressive

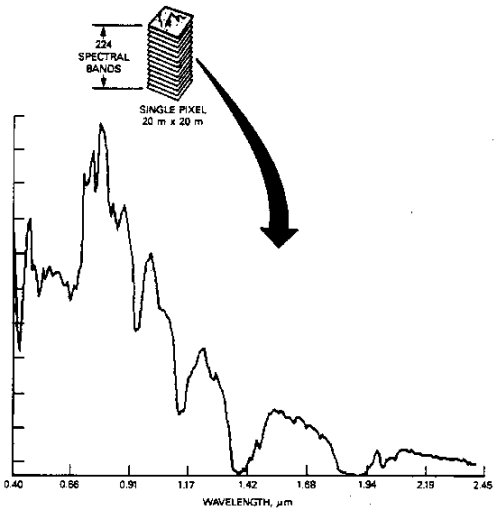


Figure 4.6: The spectral properties for Moffet image

coders. Image with high frequency content is the most suitable with the 3D SPECK coder. Since listless implementation uses a zeroblock structure for partitioning, the algorithms were also tested for hyperspectral/multispectral images and the results are shown in Table 4.2. The multispectral images were cropped to square region of 256×256 pixels using 32 bands and then transformed using the CDF-9/7 irreversible transform with five level wavelet decompositions at each dimension. For lossy compression of multispectral images, there is a slight reduction in the PSNR performance of the proposed algorithm. The measured differences are between 0.1 and 1.5 dB at certain rates. Despite the noticeable difference in PSNR, the degradation of the compressed image is not obvious because of its high frequency nature. An example of one band reconstructed using the two algorithms is shown in Figure 4.7. As an approximation, the 3D SPECK outperforms the proposed listless coder at most rates for multispectral images. However, the memory used in 3D SPECK will increase as the bit rate increase. Whereas the implementation of listless coder will be beneficial in terms of faster coding and decoding time due to fixed and reduced memory used during the coding process.

Table 4.2: Lossy compression of multispectal images

Image	Bit rate	3D SPECK	Proposed
Terrain	0.1	36.0898	32.5056
	0.25	41.3538	38.6554
	0.5	45.3238	42.792
	1	49.9873	47.5189
Urban	0.1	36.3704	32.3236
	0.25	41.1043	38.236
	0.5	44.5714	42.7114
	1	49.2803	47.1675
Lunar Lake	0.1	44.445	43.7254
	0.25	49.7817	48.8619
	0.5	52.8588	52.0911
	1	56.3312	55.2982
Moffet	0.1	40.5336	36.2321
	0.25	47.6069	43.766
	0.5	52.7854	50.2166
	1	57.9943	55.6486
Cuprite	0.1	40.5336	36.2321
	0.25	47.6069	43.766
	0.5	52.7854	50.2166
	1	57.9943	55.6486
Jasper	0.1	44.9641	44.6846
	0.25	48.3513	47.9177
	0.5	50.9033	50.3067
	1	54.8536	53.9004

4.4.3 Comparison of lossless compression performance

Progressive-lossless compression, is usually used in medical image application. Therefore, the performance of progressive-lossless compression was tested using volumetric medical images. Volumetric medical images consist of the 3D grid where voxels (volumetric pixel) contain RGB or greyscale information (in this study only greyscale information). 3D medical images consist of a group of frames/slices that represent cross sections of a part of the human anatomy. Medical images also have large storage requirements, so an efficient data compression is required to reduce the size of representation of digital data without signifi-

(a) 3D SPECK (b) 3D listless

Figure 4.7: Reconstructed Urban image band 1 at bit rate 0.1

cantly degradation or loss of quality. Lossy compression methods can reduce large amount of storage required, but the loss of visual quality might be significant. For lossless compression however, the compression ratio will be limited to between 2:1 and 3:1 but with no loss of quality. Both type of lossy and lossless compression were implemented in this work for progressive-lossless compression using 3D listless SPECK algorithms.

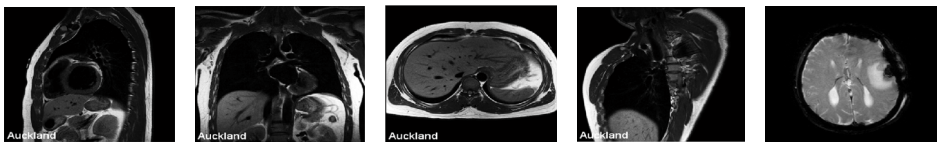
Integer wavelet transform

Selected volumetric medical images were applied with five levels of wavelet decomposition in spatial dimension and one level in the third dimension (slices/frame). Volumetric images are available from magnetic resonance imaging (MRI) scans [73] and Digital Imaging and Communications in Medicine (DICOM) images format [77]. Implementation using the 5/3 wavelet filter was based on standard JPEG2000 specifications for lossless images. The sequential plus prediction (S+P) is another low complexity multiresolution transform based on sequential and linear predictive coding. The S+P maps integers to integers, and has been used in the lossless compression of images giving better performance than the JPEG standard [78]. Table 4.3 shows the predictive weights of the S+P reversible transform developed by Said and Pearlman[78]. According to the original S+P implementation [78], Predictor B is indicated for natural, and C for medical images, and therefore only predictor C is used for this test. Figure 4.8 shows the

Table 4.3: Predictive weight of S+P transform

type	Predictive weight			
	α_{-1}	α_0	α_1	β_1
A	0	1/4	1/4	0
B	0	2/8	2/8	2/8
C	1/16	4/16	8/16	6/16

original slices of the volumetric images used in the experiment. The quality of lossy reconstruction during progressive transmission at specific bit rates is presented in Table 4.4. Using the reversible transform, lossy reconstruction using



(a) Saggital (b) Coronal (c) Tranverse (d) Short (e) Brain

Figure 4.8: Original 3D source (Slice image)

the listless coder shows a significantly better performance compared with the 3D SPECK. The improvement of over 2 dB at certain rates suggest that the operator can view a better image approximation during browsing. The results also show that although the sequential transform was used with a predictor specifically generated for medical images, the more common 5/3 tap filter outperforms S+P transform. Lossless compression performance was then measured using the final rate or bpp required to reconstruct the error free image. The results are shown in Table 4.5.

For lossless compression, the 3D SPECK outperforms the 3D listless coder in most cases in term of final bit rate to reconstruct the image. 3D SPECK require less bit to reconstruct the lossless image when compared with 3D listless coder. In the lossless case however, the S+P transform perform better than the 5/3 filter. This highlights the efficiency of the S+P transform in lossless applications. Despite lower performance using the listless algorithms, efficiency is maximised by using the proposed transform. This is due to the fact that the fixed and reduced memory used in listless implementation provides huge savings compared

Table 4.4: Lossy reconstruction of reversible transform

	CDF 5/3			S+P		
Bit rate	3D	3D listless	Δ	3D SPECK	3D listless	Δ
Coronal						
0.125	22.0821	22.8413	0.7592	19.4259	23.0703	3.6444
0.25	25.6671	27.5288	1.8617	23.6193	24.4302	0.8109
0.5	30.9173	32.0097	1.0924	28.671	29.6179	0.9469
1	37.23495	37.6137	0.37875	34.298	34.6707	0.3727
Saggital						
0.125	22.84553	24.1081	1.26257	21.5029	25.0586	3.5557
0.25	28.13142	28.8096	0.67818	25.3157	27.0517	1.736
0.5	33.21846	33.6624	0.44394	30.002	32.0321	2.0301
1	40.93693	41.7854	0.84847	37.9973	39.1153	1.118
Tranverse						
0.125	18.608	18.5365	-0.0715	16.8633	19.0679	2.2046
0.25	22.4461	22.6912	0.2451	20.0495	20.1796	0.1301
0.5	27.2393	27.6452	0.4059	24.3752	24.718	0.3428
1	32.5378	32.5781	0.0403	29.6963	29.8852	0.1889
2	40.0297	39.6733	-0.3564	19.4633	20.3129	0.8496
Short						
0.125	21.4902	22.6674	1.1772	22.3428	24.2468	1.904
0.25	24.4776	23.3627	-1.1149	24.9901	24.963	-0.0271
0.5	27.796	27.8348	0.0388	29.6841	29.7678	0.0837
1	32.479	32.4989	0.0199	19.4259	23.0703	3.6444

Table 4.5: Final bit rate of lossless compression

IWT	5/3		S+P	
Image	3D SPECK	Proposed	3D SPECK	Proposed
MRI	2.8424	2.8638	2.83464	2.8592
Coronal	2.6948	2.7614	2.6722	2.7378
Saggital	2.1915	2.2284	2.1666	2.2035
Tranverse	3.9	4.0737	3.9128	4.0196
Short	4.3009	4.4634	4.2435	4.3990
Brain	5.3539	3.4178	5.3078	3.8918

to the 3D SPECK. To prove this, the percentage saving are shown in Table 4.6. The equation to measure the memory needed is represented in Equations 4.3 and 4.4. Notice that the savings varies from image to image. The reason for this is

Table 4.6: Memory used (MegaBytes) and its saving in lossless application

Image	3D SPECK	Proposed	Saving%
MRI	0.3060	0.125	59.1537
Coronal	1.4003	0.5	64.2926
Saggital	1.2017	0.5	58.3922
Tranverse	0.9538	0.25	73.7882
Short	2.2949	0.5	78.2122
Brain	3.3821	0.6250	81.5204

that complex images will need more lists to represent location and hence greater savings can be achieved using listless coder with fixed memory.

4.5 Conclusions

The 3D listless SPECK is a new, efficient and lower memory 3D image coding method. The significant reordering/mapping of 3D wavelet coefficients ensures that it codes more significant information belonging to the lower frequency bands at an earlier stage. After that, the partitioning process based on zeroblock partitioning is similar to that of the 3D SPECK, but without any lists being required. The bit-stream generated by the proposed coder has the ability to effect progressive transmission, but produces a different level of performance depending on factors such as image type, the type of transform used and the mode of compression. Overall, the performance of the proposed listless 3D coder in terms of PSNR is lower than the 3D SPECK with multispectral and lossless volumetric medical image. However, similar or better performance can be seen in the irreversible lossy compression of medical and video images. However, the main contribution of the research described in this chapter is the implementation of the 3D listless structure, which reduces the amount of working memory required compared with the 3D SPECK whose lists that use adaptive memory. This significantly improve the efficiency of the progressive coder, with obvious salience for applications that require progressive lossless compression where the massive savings in memory required will improve the performance of progressive lossless coders.

Chapter 5

Beyond Wavelet - The Directional Filter Bank

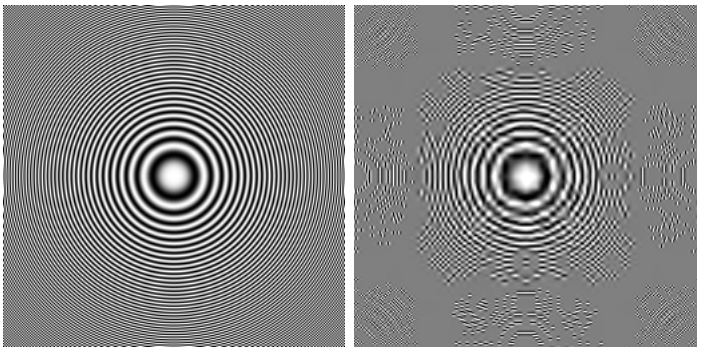
5.1 Introduction

The wavelet transform is widely used in image compression algorithms due to its decorrelation properties. Wavelet based compression offers high coding efficiency, multiresolution image representation, and acceptable level of computational complexity. Wavelet-based image and video compression has gained increasing recognition, especially since the breakthrough of the development of embedded zerotree coding algorithms. However, this has not eliminated the need for more efficient image transforms. Recent research has highlighted the fact that wavelets are unable to represent natural images efficiently due to singularities in the form of regular edges. In other words, the wavelet is not optimal in capturing two dimensional singularities such as edges and smooth curves especially in natural images. As a consequence, transforms with direction sensitivity are now being the subject of research leading to the development of geometrical image transforms. While existing geometrical transforms structures have already managed to outperformed wavelets both theoretically and in practice, at least in certain respects, consistent improvements in performance over wavelets in particular domains of image compression have not yet been achieved. This chapter first highlight the limitations of wavelet in representing smooth contours and edges in images before

going on to explore the currently available approach known as the geometrical image transform. After analysing the advantage and disadvantage of each approach, a slightly different implementation of the directional filter bank (DFB) in the wavelet domain is proposed which could improve the performance of a direction-based transform for image compression.

5.2 The limitations of wavelet and the new approaches

Over the last decade, wavelets have had a growing impact on signal processing theory and practice due to their decorrelation properties and successful use in compression applications [35]. The crux of wavelet-based algorithms is the filter banks which have become standard in signal processing operators. A good filter bank is able to represent signal approximation with few coefficients. However, as previously mentioned, wavelet transform is not optimal for the representation of images consisting of different regions of smoothly varying grey value separated by smooth boundaries. This is due to the two dimensional tensor product in wavelet bases. This problem can be seen visually in Figure 5.1, in the form of



(a) Original Zoneplate (b) Wavelet: 2048 bit

Figure 5.1: Example of compression artefact in wavelet

compression artefacts which can also be derived theoretically from the quantitative measurement using non-linear approximation(NLA) error [79].
Non-linear approximation error is a measure of rate distortion in transform coders,

and it can be computed by separating the coefficients quantised to zero from all others [80]. The resulting distortion rate depends crucially on the precision of NLA with few non-zero basis coefficients. The NLA function (also known as best M-term approximation) for a signal f, where $f = \sum_{n=1}^{\infty} c_n \phi_n$ can be defined as:

$$\widehat{f}_M = \sum_{n \epsilon I_M} c_n \phi_n \tag{5.1}$$

where I_M is the set of indexes of the M largest $|c_n|$. The quality of the approximated function $\widehat{f}_M$ relates to how sparse the expansion by $\{\phi_n\}_{n=1}^{\infty}$ is, or how well the expansion compacts the energy of f into a few coefficients [15]. From this class of function, the best M-term approximation error is defined as :

$$\|f - \widehat{f}_M\| \tag{5.2}$$

which indicates the decay rate of any basis. NLA serves as a first indicator to any transform performance in compression application. However, it must be stressed that although a powerful M-term approximation is desirable, it must be followed by an appropriate compression or quantisation technique. For a two dimensional wavelet, the basis is obtained by a tensor product of one dimensional wavelets. The M-term NLA of a simple piecewise constant function with a linear discontinuity leads to a quadratic error of the order of [9]:

$$\tilde{\varepsilon_M} \sim \frac{1}{M} \tag{5.3}$$

This rate is due to the fact that the discontinuities in images yield many wavelet coefficients of large magnitude. So the regularity over the edges remains unseen from the discrete wavelet transform. As a solution, a transforms and basis that includes some 'geometry' form providing truly two dimensional representation is needed [9]. Geometry image representation can be described as in Figure 5.2.

Next, the geometry transforms, such as the contourlet [15], curvelet [10, 11] and bandelet [12, 81, 13] are investigated. These have been developed to overcome the limitation of wavelet in providing directionality and anisotropy information. At the same time the transforms maintain beneficial properties such as multires-

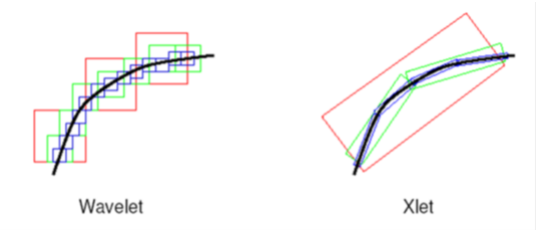


Figure 5.2: Wavelet versus the new scheme: fewer refinements in the new scheme at contours or edges

olution and localization. The approximation theory [80], related to NLA error produced by different transforms is also discussed.

5.2.1 Curvelet

Curvelet basis provides algorithmic solution to approximate geometric image where contours are C^2 [10, 11]. Approximation theory, NLA in curvelet is satisfied by the equation below:

$$\|f - f_M\|^2 = O(M^{-2}\,(logM)^3) \tag{5.4}$$

The curvelet basis is a priori fixed and the thresholding of the curvelet coefficients is sufficient to adapt the approximation to the geometry of the image. However, it has the disadvantage that the approximation is only optimal for piecewise C^α functions with $\alpha = 2$, but it is no longer optimal for $\alpha > 2$ or for less regular functions such as bounded variations function [14]. Curvelet are not practically suitable for image compression because it doesn't have orthogonal basis and the redundancy factor of the transform. Redundancy means the transform generate numbers of coefficients larger than the size of original signal, hence more significant bit generated during coding. However, application such as denoising would benefited from the efficiency of curvelet transforms to represent contours and edges.

5.2.2 Bandelet

A bandelet is a another image representation method, which can efficiently indicate the local directions and is adapted to the image's geometry [82]. It provides adaptive geometric representations that modify the image representation using geometry computed from the image [12], where a geometric flow of vectors is defined to represent the edges in an image. These vectors give the local directions in which the image has regular variations. Orthogonal bandelet bases are subsequently constructed by dividing the image support in regions where the geometric flow is parallel. Bandelets can efficiently represent image regularity and perform better than wavelet transforms. However, due to the adaptive nature of this transform it requires the detection of the discontinuity curve beforehand, so that more computation is required compared to other fixed transforms such as contourlets and wavelets.

5.2.3 Contourlet

The contourlet transform which was developed by Min Do and Vertelli [15] is among the most significant, because it was specifically developed for digital images in discrete domains and provides efficient directional multiresolution expansion. However the contourlet is a redundant transform due to the implementation of the Laplacian pyramidal filter. Redundancy means that coding process requires more coefficients to be coded, which will be likely to reduce the coding performance. However, it has been proven that contourlets perform very well compared to wavelets for low bit rate coding [83]. The objectives of the development of contourlet transforms are [15]:

- Multiresolution representation: enabling images to be successfully approximated, from coarse to fine resolutions.
- Localisation: the basis elements should be localized in spatial and frequency domain.
- Critical sampling : to reduce redundancy.
- Directionality: containing basis elements oriented at multidirection.
- Anisotropy: to capture smooth contours in images by using basis elements with variety of elongated shapes with different aspect ratios.

A contourlet consists of two main blocks: the Laplacian pyramid and directional filter bank. The directional filter bank was designed to capture high frequency information representing directional information from the input image. As a result, low frequency content is poorly handled. Contourlets address this issue by combining the multiscale decompositions of Laplacian pyramidal filters in order to remove low frequencies in the input image before applying the directional filter bank.

5.2.4 Wavelet based contourlet transform

The wavelet-based contourlet transform [16] has been proposed to remove redundancy in contourlet transform and to make it suitable for image compression. The wavelet-based contourlet transform (WBCT) and its set partitioning in hierarchical trees (SPIHT) coding is described in [84]. A study of SPECK-based coding with WBCT shows that block based coding outperform the SPIHT-based WBCT [85]. This is due to set partitioned embedded block coder (SPECK)'s ability to capture more high frequency information from images with a lot of textural details. The no-list SPIHT(NLS) has also been combined with the WBCT [18] showing improvements of about 0.2dB at low bit rates. However, WBCT generates the artefact called the pseudo-Gibbs phenomena on compressed images; when some of the transforms are set to zero during NLA. The ringing artefact is obvious for DFB and the issue is addressed in contourlets by applying DFB to the Laplacian pyramid, where a highpass channel is free from frequency scrambling [15]. This artefact can be quite obvious especially in smooth images like Lenna and the distortion rate of the WBCT can be quite low due to this artefact. A combination of WBCT at certain decomposition levels and normal wavelets has been proposed to improve the performance, leading to the introduction of the hybrid wavelet and directional filter banks (HWD) family [17]. The HWD offers better performance than contourlets, in terms of peak signal-to-noise ratio (PSNR) during NLA reconstruction, but the resulted image coding still suffers the pseudo-Gibbs phenomenon.

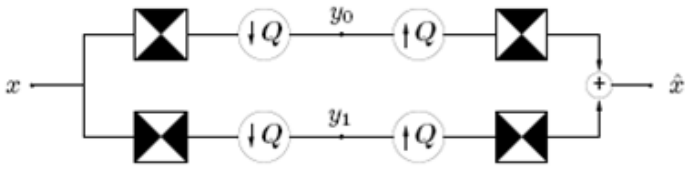
5.2.5 Directional filter bank for image decomposition

The directional filter bank used for image decomposition consists of the separation of images into several components, each containing directional information in some specific directions. Decomposition into directional components has successfully been used for feature extraction and pattern recognition [86, 87]. The first two dimensional DFB was developed by Bamberger and Smith, and it can be maximally decimated while achieving perfect reconstruction [86, 88, 89, 90]. The DFB was implemented in a tree structure composed of a two band system, first by modulating the signal and filter with a diamond shaped filter [88]. The second step involves parallelogram shaped characteristic passband to resample the matrices. In the contourlet transform, construction of the DFB used a simpler approach based on two building blocks consisting of two-channel quincunx filter banks with fan filter as shown in Figure 5.3(a), followed by a shearing operator in Figure 5.3(b), to re-order the image samples [15]. This approach provides different directional frequency partitions while maintaining perfect reconstruction properties. The key to the DFB is to achieve the desired spectrum division (Figure 5.3(c)) using an appropriate combination of the shearing operation with the two directional partition of quincunx filter banks at each node in a binary tree-structured filter bank. Filter bank implementation in the DFB is based on a ladder structure [91].

5.3 Hybrid wavelet and directional filter bank

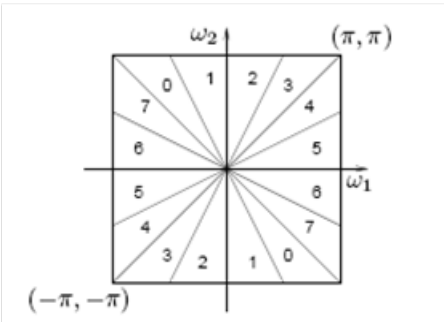
5.3.1 Construction

It was proposed in this study to improve the directionality of wavelet transform by employing the directional filter bank onto the highpass channel in the wavelet domain. However, unlike in the previous implementations, the DFB is only implemented onto the highpass band (HH) in wavelet decompositions. The directional filter used is based on contourlet transforms. Therefore, the term wavelet-based contourlet in the highpass domain (WBChh) was coined to represent the proposed filter. One of the major drawbacks of using the HWD [16, 17, 92] and WBCT [18, 84, 85] is the pseudo-Gibbs artefacts introduced when some of the



(a) First block: quincunx filter

(b) Shearing operation



(c) Frequency partitioning

Figure 5.3: DFB in contourlet

transform coefficients are set to zero during NLA and coding. In both methods, the artefacts become severe with the implementation of DFB in discrete wavelet transform (DWT) because the long filter is used in DFB for better directional resolutions in directional basis functions such as wavelet [17]. This is the main reason for the application of contourlet transforms to the directional filter bank in Laplacian pyramid in which the highpass channel is free from frequency scrambling [15]. The objectives of this work is to find a balanced solution between maintaining compression performance and improving directionality by:

1. Providing non redundant transform for image compression coding;
2. Avoiding frequency scrambling when using DFB in DWT;
3. Only implementing DFB in the finer levels to remove the pseudo-Gibbs artefact
4. A transform with directionality at lower computational complexity

The wavelet as a directional basis

As previously described in Section 2.4, a two dimensional wavelet transform constructed from one dimensional scaling function, $\phi(t)$ and wavelet function, $\omega(t)$, and forms a tight frame of an orthonormal basis $L^2(\mathbb{R})$. The decomposition of multiresolution space representation in Equation 2.14 correspond to [22]:

$$V_{j-1} \otimes V_{j-1} = (V_j \otimes V_j) \oplus (V_j \otimes W_j) \oplus (W_j \otimes V_j) \oplus (W_j \otimes W_j) \tag{5.5}$$

where $(V_j \otimes V_j)$ is the coarse scale approximation, while one has the following horizontal, vertical, and diagonal detail spaces:

$$W_j^H = V_j \otimes W_j,\ W_j^V == W_j \otimes V_j,\ W_j^D = W_j \otimes W_j. \tag{5.6}$$

Each of the three wavelet spaces is spanned with a wavelet, where the mother wavelets are:

$$\begin{aligned} \omega^H(t) &= \phi(t_1)\,\omega(t_2) \\ \omega^V(t) &= \omega(t_1)\,\phi(t_2) \\ \omega^D(t) &= \omega(t_1)\,\omega(t_2) \end{aligned} \tag{5.7}$$

This type of decomposition is known as an isotropic wavelet decomposition, and are often used in image processing, since it has a square support that not stretched along the axes. Two dimensional isotropic wavelet correspond to a non-separable basis, and require three mother wavelets in Equation 5.7. Figure 5.4[22] shows an examples of this wavelets. One level decomposition of two dimensional isotropic wavelet is shown in 5.5.

Frequency scrambling occurs in the wavelet filter bank because after downsampling, a highpass channel is folded back into the low frequency band (Figure 5.5), and thus its spectrum is reflected. The regions which contain frequency scrambling are horizontal and vertical wavelet subbands, so directional decomposition is only applied to the diagonal wavelet subband in order to avoid the low frequency regions where frequency scrambling problems arise. This is the

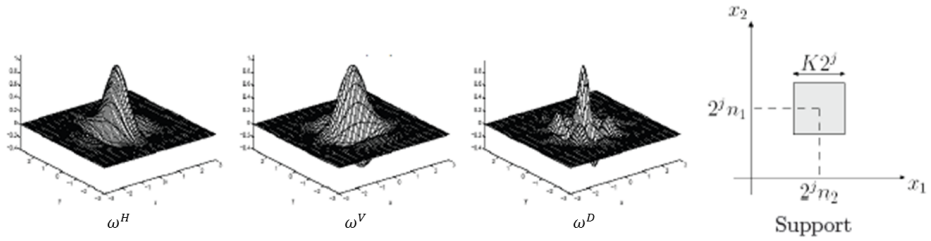


Figure 5.4: The 2D isotropic wavelet and the approximative support

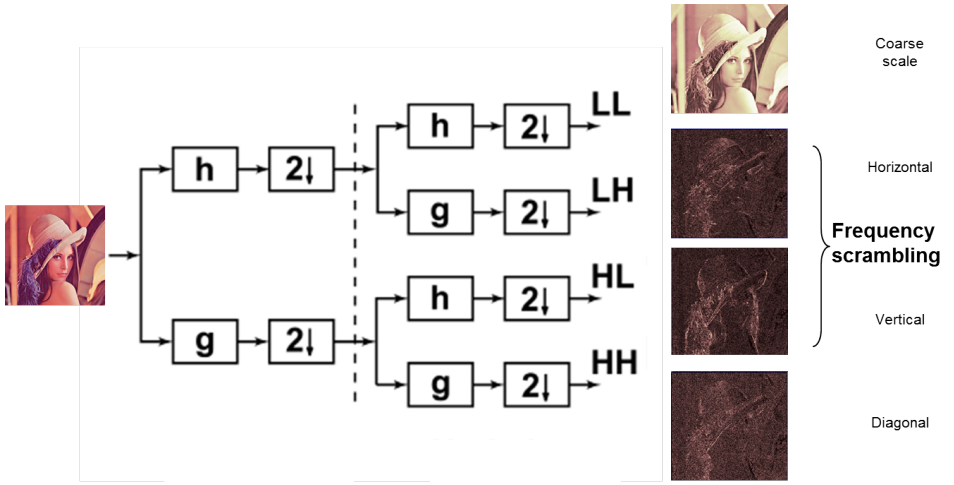


Figure 5.5: One level 2D isotropic wavelet decomposition

simplest and yet efficient way to introduce directionality into wavelet transforms of an images and at the same time avoid frequency scrambling.

5.3.2 Computational complexity

This section investigates the computational complexity of the proposed transforms. The discussion include an analysis of complexity in wavelets, contourlets and the HWD. The proposed transform uses the ladder structure, the pkva filter [15] used in contourlets. The DWT and the DFB are two independent stages similar to the Laplacian pyramid filter and the DFB in original contourlets [15]. Thus, the calculation of computational complexities is by adding the complexity of each stage to produce the total complexity of one level of transform. The computational complexity measurement is based on the number of real multiplications

(R_M) and real additions (R_A) required for each input sample. Non-redundant systems such as the wavelet transform, have critical sampling which produces the number of transform coefficients equal to the number of input samples or the original image size. Consequently, in this case, computational complexity can be represented as the multiplications per input sample (MPS) and additions per input sample (APS). All transform using iteration of an elementary one level transforms. For example, in a one level transform that requires C operations per input sample, then the total complexity for the L level is given by equation [93]:

$$C_{total} = C + \frac{C}{4} + \frac{C}{4^2} ... + \frac{C}{4^{L-1}} < \frac{4C}{3} \tag{5.8}$$

Wavelet complexity

First, the complexity of wavelet transforms is considered. The CDF-9/7 filter bank decomposition can be constructed by cascading the filter in two dimensions. This is represented in Figure 5.6. Accordingly, for a one dimensional filter the

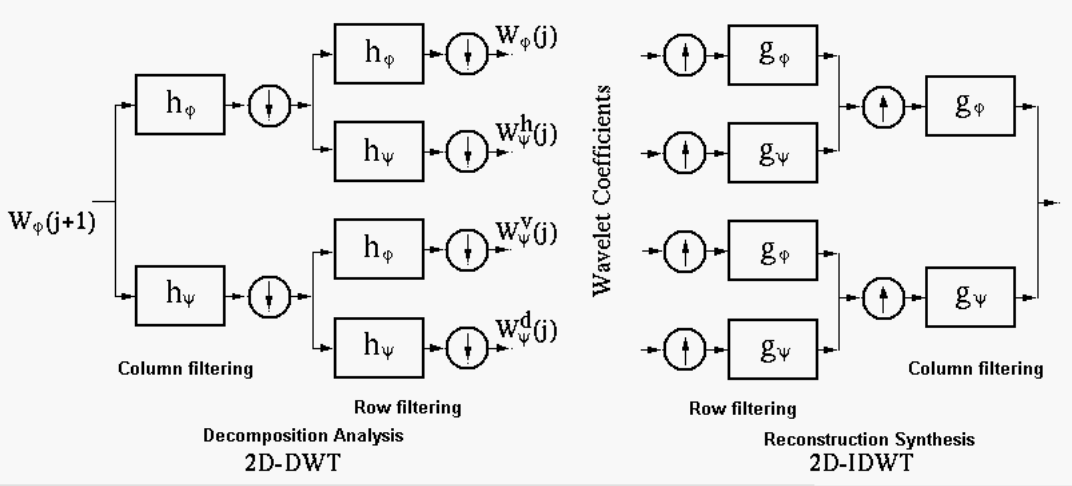


Figure 5.6: Cascade decomposition

input is lowpass and highpass filtered and decimated to generate two decomposed subbands. Decimation is a resampling process, and so it does not add to the complexity. Filtering requires convolution between the input signal and the filter.

For a filter with L length the convolution requires:

$$\begin{aligned} R_M &= L \\ R_A &= L - 1 \end{aligned} \tag{5.9}$$

So, for each input sample using one dimensional CDF-9/7 tap filter, a 9 tap lowpass filter takes 9 real multiplications and 8 real additions. A 7 tap highpass filter uses 7 real multiplications and 6 additions. Using two analysis filters gives a total of 16 MPS and 14 APS. However, the decimation process that follows filtering discards half of the samples, and therefore the number of operations for each input sample are:

$$\begin{aligned} R_M &= \frac{16}{2} = 8MPS \\ R_A &= \frac{14}{2} = 7APS \end{aligned} \tag{5.10}$$

For a two dimensional wavelet with image size, $N_1 \times N_2$, the cascading filter requires:

$$\begin{aligned} R_M &= \frac{N_1 \cdot N_2 \cdot 8 + N_2 \cdot N_1 \cdot 8}{N_2 N_1} = 16MPS \\ R_A &= \frac{N_1 \cdot N_2 \cdot 7 + N_2 \cdot N_1 \cdot 7}{N_2 N_1} = 14APS \end{aligned} \tag{5.11}$$

Based on the Equation 5.8, the total complexity of wavelet transform with five level decompositions is therefore:

$$Multiplications = \frac{4 \times 16}{3} = 21.333MPS \tag{5.12}$$

$$Additions = \frac{4 \times 14}{3} = 18.667APS \tag{5.13}$$

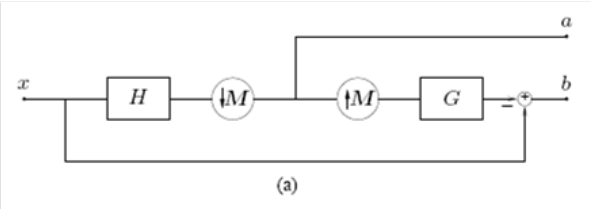


Figure 5.7: Laplacian decomposition

HWD complexity

The total complexity of the HWD, in this case is HWD-F [17] where the directional filter is applied on each subband of wavelet decomposition is given as:

$$\frac{4}{3}l_h + 3\left(3\left(\frac{l_r}{2}\right)+1\right)\sum_{j=1}^{J_m}\left(\frac{l_j}{4^j}\right)MPS \tag{5.14}$$

$$\frac{4}{3}l_h + 9\left(\left(\frac{l_r}{2}\right)-\frac{1}{2}\right)\sum_{j=1}^{J_m}\left(\frac{l_j}{4^j}\right)APS \tag{5.15}$$

where,
l_r is the length of the *quincunx filter bank*, QFB,
l_h denotes the length of the wavelet filter,
l_j denotes the DFB level ($1 \leq j \leq Jm$).
So, the total complexity of using the CDF-9/7 wavelet and triplet halfband filter in HWD-F with $l_1 = l_2 = 3$ directional level and directional level, $l_j = 2$ and the number of direction, $l_r = 8$ are:

$$Multiplications = \frac{4\times 16}{3} + 3\left(3\left(\frac{8}{2}\right)+1\right)\left(\frac{3}{4^1}+\frac{3}{4^2}\right) = 57.8955MPS \tag{5.16}$$

$$Additions = \frac{4\times 14}{3} + 9\left(\left(\frac{8}{2}\right)-\frac{1}{2}\right)\left(\frac{3}{4^1}+\frac{3}{4^2}\right) = 48.1979APS \tag{5.17}$$

Contourlet complexity

The measurement of contourlet complexity for N pixel is represented with pyramidal decomposition and the DFB respectively [15]:

$$\sum_{j=1}^{J} N \left(\frac{1}{4}\right)^{j-1} \left(\frac{L_p}{2} + 1\right) \tag{5.18}$$

$$\sum_{j=1}^{J} N \left(\frac{1}{4}\right)^{j-1} L_d l_j \tag{5.19}$$

where, j denotes the level of decompositions,
N for the number of pixel in image,
L_p denotes the length of the pyramid filter,
L_d denotes the length of the directional filter,
l_j denotes the DFB level.
The computational complexity is added to produce the total complexity of a one level contourlet, because the transforms consist of two independent stages. For one level Laplacian pyramid decompositions as in Figure 5.7 that uses CDF-9/7 filter, the Equation 5.18 is considered. With CDF-9/7 filter, the lowpass filter requires $\frac{9}{2} + \frac{9}{4} = 6.75$ MPS and $\frac{8}{2} + \frac{8}{4} = 6$ APS. Then the approximation is upsampled and filtered with the 7 tap lowpass filter to produce the predictive version of the original image, which requires $\frac{7}{2} + \frac{7}{4} = 5.25$ MPS and $\frac{6}{2} + \frac{6}{4} = 4.5$ APS. After that, one addition is required to calculate the difference between the original and predict signal. As a result, based on Equation 5.18, the one level Laplacian pyramid requires:

$$R_M = 6.75 + 5.25 = 12MPS \tag{5.20}$$

$$R_A = 6 + 4.5 + 1 = 11.5APS \tag{5.21}$$

For the DFB decomposition as in Equation 5.19, the ladder structure shown in Figure 5.8 [91] is used to construct the DFB pkva filter. $\beta(z)$ in the figure is the filter with length L, which is a cascade of a one dimensional filter in two dimensions. So, the complexity for using pkva filter with the length of 12, two-dimensional transform requires $2L = 24$ multiplications and $2(L-1) = 23$

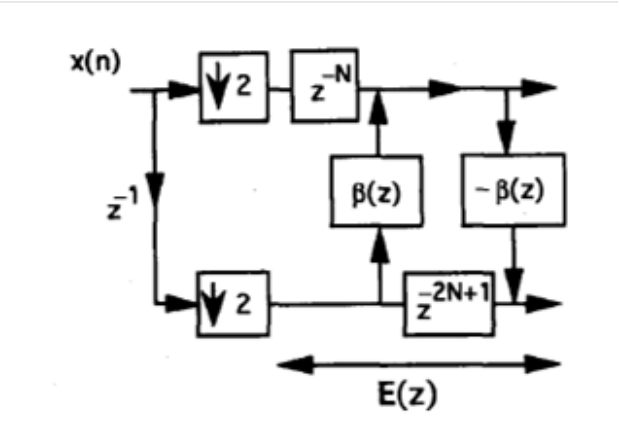


Figure 5.8: Ladder structure

additions. With 2^k band of direction in the DFB, where k stage three of filter banks, complexity of the DFB based on Equation 5.19 are :

$$R_M = 24\,(k)\,MPS \tag{5.22}$$

$$R_A = 23\,(k)\,APS \tag{5.23}$$

This gives the total complexity of the one-level contourlet transforms as [94] :

$$R_M\,(k) = 12 + 24\,(k)\,MPS \tag{5.24}$$

$$R_A\,(k) = 11.5 + 23\,(k)\,APS \tag{5.25}$$

For contourlet transform with directional level $\{l_j\}_{5>j>1} = \{0, 0, 0, 4, 5\}$ where l_1 is the finest decompositions requires :

$$R_M = \frac{4 \cdot 12}{3} + \frac{1}{4^2}\,(24 \cdot 4) + \frac{1}{4^1}\,(24 \cdot 5) = 52MPS \tag{5.26}$$

$$R_A = \frac{4 \cdot 11.5}{3} + \frac{1}{4^2}\,(23 \cdot 4) + \frac{1}{4^1}\,(23 \cdot 5) = 49.8333APS \tag{5.27}$$

WBChh complexity

For the DFB, with l level full binary tree decomposition the complexity of the DFB is multiplied by l. This holds true because the initial decomposition block in the DFB is followed by two blocks at half rate, four blocks at quarter rate and so

on [15]. The complexity of the directional filter within the wavelet decomposition is:

$$L_d \sum_{j=1}^{J} \left(\frac{1}{4}\right) 12 l_j \tag{5.28}$$

where l_j is the DFB level decompositions to the j highpass channel of the DWT, and L_d is the number of filters used in the DFB. If L_w is the number of filters used in the wavelet decompositions, the complexity of WBChh is a combination of both transforms:

$$\frac{4}{3} L_w + L_d \sum_{j=1}^{J} \left(\frac{l_j}{4^j}\right) \tag{5.29}$$

The DFB filter used is a pkva filter with length 12 [91] (ladder filter) from the contourlet transform, so for five level wavelet decompositions with directional level $\{l_j\}_{5>j>1} = \{0, 0, 0, 3, 3\}$ requires :

$$R_M = \frac{4 \cdot 16}{3} + \frac{1}{4^2}(24 \cdot 3) + \frac{1}{4^1}(24 \cdot 3) = 43.8333 MPS \tag{5.30}$$

$$R_A = \frac{4 \cdot 14}{3} + \frac{1}{4^2}(23 \cdot 3) + \frac{1}{4^1}(23 \cdot 3) = 40.2291 APS \tag{5.31}$$

From the complexity calculations, additional directional information significantly adds to the complexity of the wavelet decompositions. However, the proposed transform, WBChh has lower complexity compared to HWD and the contourlet transform. In summary, the complexity of the transform follows an increasing order as shown in Table 5.1.

Table 5.1: Comparison of computational complexity

Transform	Total Additions	Total Multiplication
Wavelet	18.667	21.333
WBChh	40.2291	43.8333
Contourlet	49.8333	52
HWD	48.1979	57.8955

5.4 Application and results

In this section, the WBChh transforms were analysed using non-linear approximation before accessing its their potential for image compression application. The performance measures used for evaluation are PSNR and structural similarity (SSIM)[25]. PSNR for greyscale images is calculated using Equations 2.6 and 2.7.

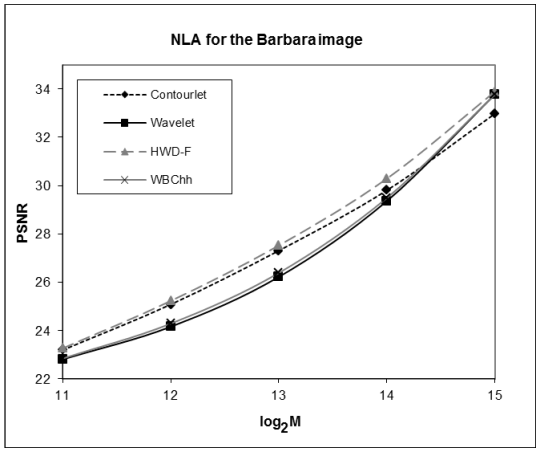
5.4.1 Non-linear approximation

Non-linear approximation(NLA) (Equation 5.32) behaviour of a scheme can be used as a first indicator of its potential for image coding. The proposed transform, WBChh was tested on various greyscale images and compared with the contourlet, wavelet and WBCT/HWD. For a fair comparison, all of the transforms used five level decompositions with Daubechies 9/7 Filter. The contourlet directional level is $\{l_j\}_{5>j>1} = \{0, 0, 0, 4, 5\}$ where l_1 is the finest decompositions [15], while the HWD-F and the proposed transform used $\{l_j\}_{5>j>1} = \{0, 0, 0, 3, 3\}$ [17]. All images were tested using the same decomposition and directional level. Equation 5.32 represent the reconstructed image,x.

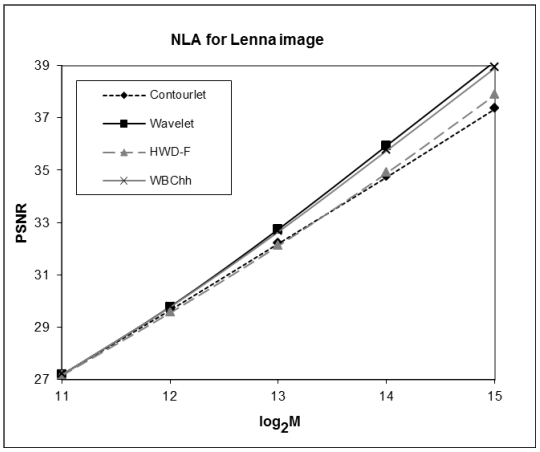
$$\widetilde{x}_M = \sum_{n \in I_M} C_n \psi_n \tag{5.32}$$

where M refers to the number of significant pixels to be kept for reconstruction. The numerical PSNR values for the NLA performance of the tested images are given in Table 5.2. The SSIM results followed in Table 5.3. Visual graphical representations of the resulting distortion rate, PSNR for NLA performance are shown in Figure 5.9 for the six tested images. Barbara, Zoneplate, Bicycle and Fingerprint were used for texture and contour, while Lenna and Peppers were smooth images.

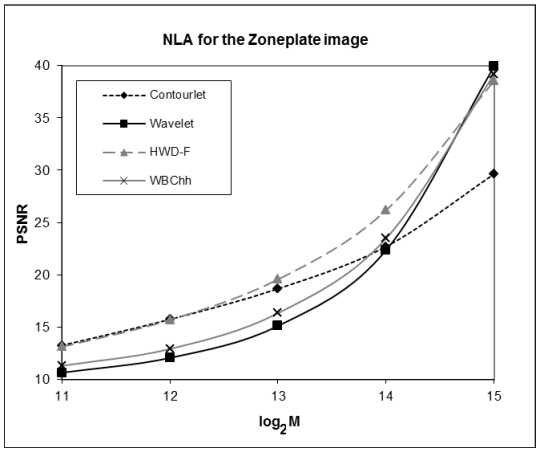
The distortion rate, PSNR results from Table 5.2 and Figure 5.9 shows that in smooth images like Lenna and Peppers, the wavelet transform achieves the best PSNR when the significant coefficients is set to $M = 4096$ or larger. Overall, at low bit rates, or where $M = 2048$ ($log_2 M = 11$ in the graph) the best



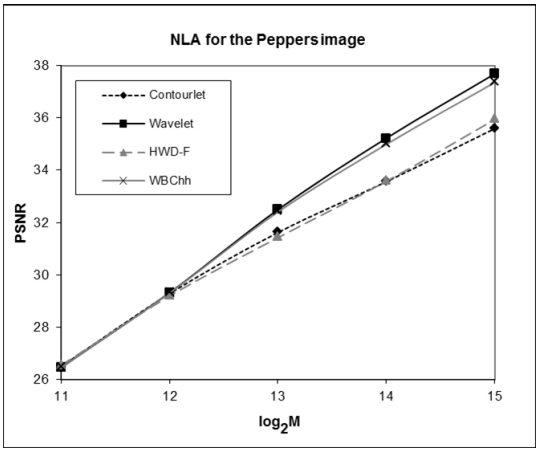
(a) Barbara Image



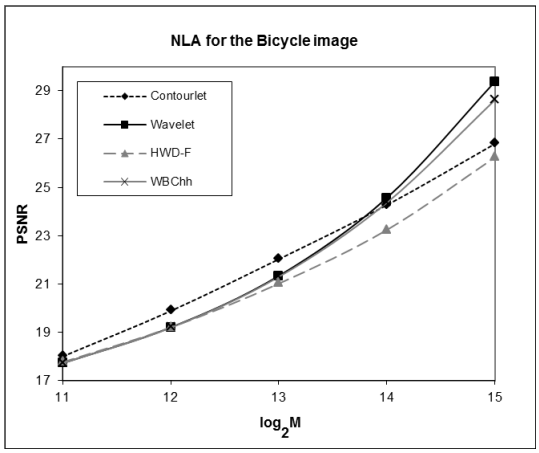
(b) Lenna Image



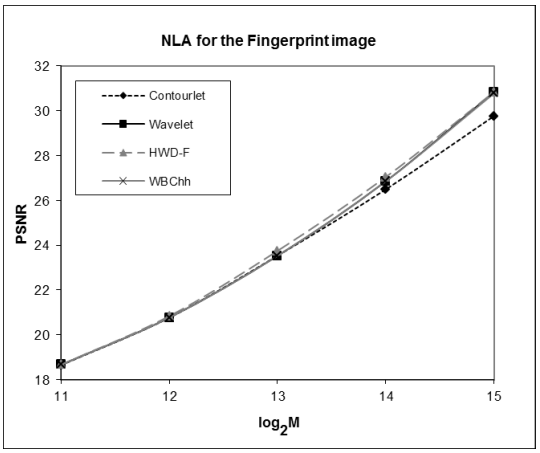
(c) Zoneplate Image



(d) Pepper Image



(e) Bicycle Image



(f) Fingerprint Image

Figure 5.9: Rate distortion result for NLA comparison

Table 5.2: PSNR values of the NLA performance

Method/M	2048	4096	8192	16384	32768
Barbara					
Contourlet[15]	23.1898	25.0784	27.3057	29.8047	32.9768
Wavelet	22.8052	24.1557	26.2184	29.3596	33.7731
HWD-F	**23.2622**	**25.2258**	**27.5204**	**30.2889**	**33.8888**
WBChh	22.8405	24.2855	26.3839	29.4878	33.7699
Lenna					
Contourlet[15]	**27.1922**	29.6501	32.1945	34.7330	37.3385
Wavelet	27.1824	**29.7819**	**32.7455**	**35.9403**	**39.1609**
HWD-F	27.1528	29.5500	32.1103	34.8895	37.8697
WBChh	27.1828	29.7690	32.6471	35.7459	38.9010
Peppers					
Contourlet[15]	26.4949	29.2846	31.6214	33.5627	35.5941
Wavelet	26.4606	**29.3200**	**32.4970**	**35.2262**	**37.6857**
HWD-F	**26.5425**	29.1925	31.4252	33.5781	35.9421
WBChh	26.4606	29.3075	32.4072	34.9924	37.3567
Zoneplate					
Contourlets[15]	13.2670	15.7514	18.6840	22.7076	29.6351
Wavelet	10.6659	12.0751	15.1312	22.3594	**39.9156**
HWD-F	**13.1231**	**15.7037**	**19.5729**	**26.1708**	38.5399
WBChh	11.3163	12.9274	16.3480	23.4868	39.1551
Bicycle					
Contourlet[15]	**17.9983**	**19.8834**	**22.0272**	24.2667	26.8046
Wavelet	17.7221	19.2014	21.3381	**24.5670**	**29.3750**
HWD-F	17.7627	19.1993	21.0146	23.2385	26.2434
WBChh	17.7212	19.1887	21.2955	24.3677	28.6178
Fingerprint					
Contourlet[15]	18.6533	20.7702	23.5359	26.4905	29.7606
Wavelet	18.6698	20.7735	23.5286	26.8531	30.8314
HWD-F	**18.6762**	**20.8385**	**23.7495**	**27.0427**	**30.8389**
WBChh	18.6698	20.7735	23.5267	26.8500	30.7773

PSNR is either the HWD-F/WBCT or the contourlet transform. In images with contours, HWD-F achieves better results than the others. This was expected, since the HWD-F/WBCT has the highest complexity. The next best results were achieved by the contourlet and the proposed transform, followed by the wavelet. The proposed transform, WBChh performed very close to wavelet in the NLA comparison.

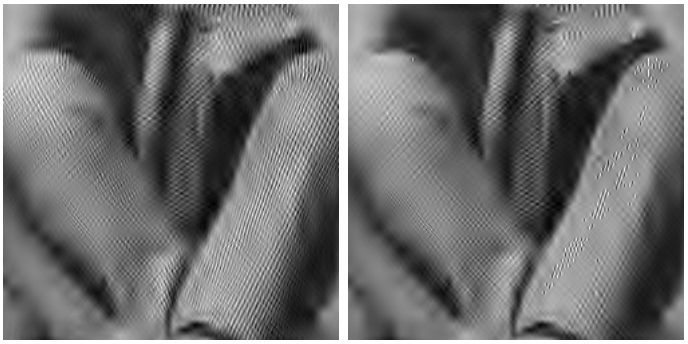
A visual quality for comparison using the Barbara image when $M = 2048$, is

Table 5.3: SSIM values of the NLA experiment

Method/M	2048	4096	8192	16384	32768
Barbara					
Contourlets [15]	0.7147	0.8041	0.8788	0.9293	0.9636
Wavelet	0.7122	0.7840	0.8549	0.9228	0.9689
HWD-F	**0.7231**	**0.8145**	**0.8824**	**0.9307**	**0.9670**
WBChh	0.7153	0.7895	0.8598	0.9242	0.9683
Lenna					
Contourlets [15]	0.8201	0.8855	0.9322	0.9611	0.9785
Wavelet	**0.8234**	**0.8888**	**0.9377**	**0.9679**	**0.9848**
HWD-F	0.8213	0.8800	0.9253	0.9579	0.9793
WBChh	0.8232	0.8887	0.9369	0.9666	0.9839
Peppers					
Contourlets [15]	0.8141	0.8889	0.9310	0.9539	0.9712
Wavelet	**0.8210**	**0.8890**	**0.9394**	**0.9661**	**0.9809**
HWD-F	0.8088	0.8790	0.9187	0.9474	0.9708
WBChh	**0.8210**	**0.8890**	0.9386	0.9643	0.9792
Zoneplate					
Contourlets [15]	0.7389	0.8746	0.9448	0.9795	0.9955
Wavelet	0.4087	0.6220	0.8460	0.9750	**0.9996**
HWD-F	**0.7528**	**0.8755**	**0.9517**	**0.9895**	0.9994
WBChh	0.5307	0.7233	0.8918	0.9807	0.9995
Bicycle					
Contourlets [15]	0.5913	**0.7257**	**0.8248**	0.8898	0.9337
Wavelet	**0.5948**	0.7161	0.8235	**0.9043**	**0.9609**
HWD-F	0.5752	0.6873	0.7826	0.8523	0.9108
WBChh	0.5946	0.7125	0.8176	0.8959	0.9537
Fingerprint					
Contourlets [15]	0.5649	0.7398	0.8757	0.9479	0.9811
Wavelet	**0.5682**	0.7409	0.8757	0.9516	0.9862
HWD-F	0.5681	**0.7440**	**0.8822**	**0.9567**	**0.9866**
WBChh	**0.5682**	0.7409	0.8755	0.9515	0.9861

shown in Figure 5.10. From the reconstructed Barbara image in Figure 5.10 at 2048 significant coefficients, it can be seen that the contourlet provides the best image with clear contour line at Barbara pants, followed by the HWD-F/WBCT, WBChh and finally the wavelet. This is as predicted since the aim was to provide a nonredundant directional transform that not only improves the edge information compared to wavelets but also maintains the distortion rate for smooth images like Lenna and Peppers using the same structure. The PSNR values from graph

(a) Original (b) contourlet (c) Wavelet



(d) HWD-F (e) WBChh

Figure 5.10: Barbara at 2048 coefficients (zoom)

(a) Original (b) contourlet (c) Wavelet

(d) HWD-F (e) WBChh

Figure 5.11: Lenna at 4096 coefficients (zoom)

shown in Figure 5.9(b) and 5.9(d) shows that the WBChh results are very close to those of the wavelet for Lenna and Peppers. The visual quality for comparison using images with smooth region is shown in zoomed Lenna image in Figure 5.11, proving the resulted PSNR in NLA test.

In conclusion, the NLA comparison shows that both the contourlet and the HWD-F/WBCT show good performance compared to wavelet and the proposed transform. In addition, the proposed transform also outperforms wavelet at the most bit rates, whereas did not outperform the HWD and contourlet. This is expected since the complexity of the HWD is the highest, due to its implementation of the directional filter in each wavelet subband. The complexity then followed by the contourlet due to the implementation of redundant Laplacian decompositions. The redundancy in the contourlet means that performance decline as bit rate increased. The proposed transform, WBChh that apply DFB only to highpass channel (HH), has the lowest complexity among the directional-based transform but higher when compared to wavelet. Despite a lower performance in the NLA comparison, wavelet transforms have the benefit of embedded coding

which efficiently encodes coefficients based on a zerotree or zeroblock structure. Clearly, the NLA comparison can be used as a first indicator of potential for image coding, but further quantization is important during actual coding. Coding performance using the proposed transform is investigated in the next section.

5.4.2 Image coding performance

While a good NLA has a desirable properties, it must be followed by appropriate compression. For example, although HWD provides better NLA compared to other transform, the performance of actual coding is still very close to wavelets [17]. In this case, the WBChh transform is coded with SPECK-based coding with listless structure(LsK) [42]. This structure eliminates the use of lists and maintains a zeroblock partitioning algorithm while performing quite well as a coding technique at level of complexity lower than those in SPIHT partitioning. Coding performance was compared against that of the wavelet SPIHT and HWD-F CSPIHT [17, 95] coders. A visual representation of CSPIHT coding used with HWD-F is shown in Figure 5.12 [16, 95].

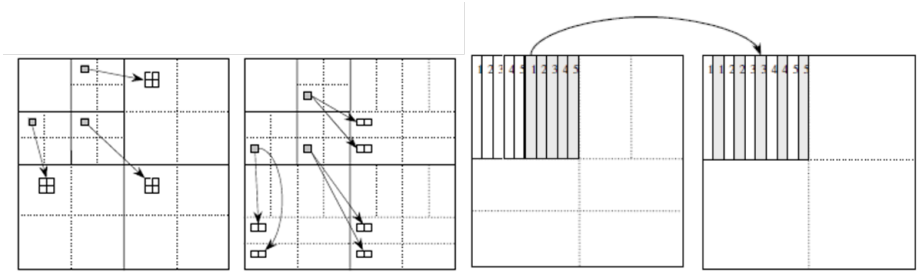


Figure 5.12: Zerotree relation used in CSPIHT HWD

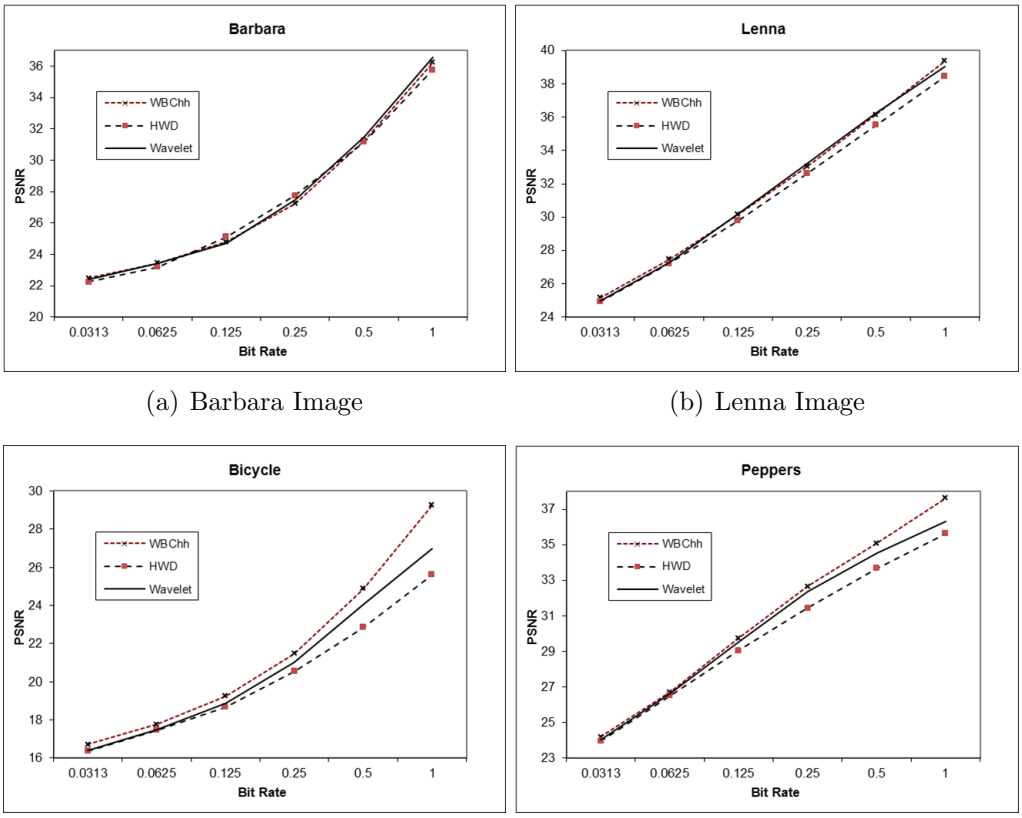
The full results follow in Tables 5.4 and 5.5. From the performance results shown in Figure 5.13, it can be seen that the PSNR of the proposed transform performs better than that of the HWD and wavelet with the Bicycle and Peppers images, but performs very closely to the wavelet with Barbara and Lenna. The proposed transform outperforms the HWD and wavelets at very low bit rates around 0.1 to 0.2 dB with the ability to retain edge information (Figure 5.14). As the bit rate progresses, the proposed transform outperforms the HWD with smooth images like Lenna and Peppers but maintains level of performance very

Table 5.4: Performance of the proposed coder without entropy coding

Greyscale Image	Bit Rate	PSNR			SSIM		
		Wavelet	HWD	WBChh	Wavelet	HWD	WBChh
Elaine	0.0313	25.7769	25.7606	**26.0442**	0.7664	0.7659	**0.7747**
	0.0625	28.2579	28.1175	**28.419**	0.848	0.8419	**0.8537**
	0.125	30.4107	30.1081	**30.4345**	0.9037	0.8954	**0.9074**
	0.25	**31.9566**	31.5892	31.9552	0.9384	0.9312	**0.9426**
	0.5	33.0985	32.6958	**33.6492**	0.9541	0.9449	**0.959**
	1	35.0536	34.9492	**35.7385**	0.9718	0.969	**0.9767**
Lenna	0.0313	24.7546	24.6828	**25.0469**	0.7365	0.7534	**0.7564**
	0.0625	27.1246	27.0671	**27.3784**	0.827	0.8379	**0.8439**
	0.125	30.0037	29.6161	**30.0485**	0.8905	0.9043	**0.9109**
	0.25	33.0606	32.374	**32.9348**	0.936	0.9472	**0.9513**
	0.5	36.1279	35.2969	**36.0554**	0.9665	0.9727	**0.9746**
	1	38.9155	38.2763	**39.2863**	0.9842	0.9879	**0.9892**
Barbara	0.0313	21.6697	21.4521	**22.4401**	0.6191	0.6528	**0.6775**
	0.0625	22.8091	22.7033	**23.3916**	0.7085	0.7229	**0.7436**
	0.125	24.1523	24.4067	**24.7116**	0.7942	0.8066	**0.8082**
	0.25	26.833	26.902	**27.2113**	0.8761	0.8844	**0.9004**
	0.5	30.5362	30.1278	**31.2047**	0.9327	0.9469	**0.9545**
	1	35.4396	34.4858	**36.1477**	0.9728	0.9819	**0.9844**
Car	0.0313	22.2743	22.0822	**22.4509**	0.6347	0.6818	**0.6942**
	0.0625	24.0587	23.8362	**24.2566**	0.7322	0.7364	**0.7626**
	0.125	26.1679	25.8865	**26.3516**	0.8256	0.8355	**0.8513**
	0.25	28.7884	28.3531	**29.0622**	0.8961	0.9069	**0.9224**
	0.5	**32.069**	31.4991	32.0173	0.9483	0.9587	**0.9617**
	1	**36.961**	35.9794	36.8232	0.9789	0.9835	**0.9855**
Baboon	0.0313	19.7749	19.7584	**19.8949**	0.425	0.4696	**0.4751**
	0.0625	20.3437	20.3278	**20.5082**	0.5287	0.5402	**0.5439**
	0.125	21.3344	21.3054	**21.4256**	0.6526	0.6787	**0.693**
	0.25	22.7266	22.6422	**22.8042**	0.7605	0.764	**0.7694**
	0.5	**24.9795**	24.7772	24.7531	0.861	0.8829	**0.8856**
	1	**28.4864**	28.0966	28.1435	0.9335	0.951	**0.9522**
Bicycle	0.0313	16.3194	16.2649	**16.6713**	0.4483	0.4926	**0.5091**
	0.0625	17.4037	17.308	**17.7085**	0.5628	0.6117	**0.6296**
	0.125	18.7807	18.5458	**19.2108**	0.6908	0.7448	**0.7484**
	0.25	20.8948	20.338	**21.3599**	0.7885	0.8465	**0.8592**
	0.5	23.8837	22.6195	**24.7703**	0.8655	0.9042	**0.9212**
	1	26.8693	25.406	**29.0502**	0.9271	0.9599	**0.9676**
Peppers	0.0313	23.8673	23.7776	**24.1148**	0.7351	0.7551	**0.7566**
	0.0625	26.4805	26.2804	**26.6493**	0.8144	0.844	**0.8477**
	0.125	29.403	28.7903	**29.6597**	0.8803	0.8897	**0.909**
	0.25	32.2013	31.2075	**32.5487**	0.9239	0.934	**0.9502**
	0.5	34.4417	33.5021	**35.0156**	0.9521	0.9617	**0.9702**
	1	36.1728	35.5253	**37.4441**	0.9716	0.9791	**0.9804**

Table 5.5: PSNR and SSIM performance of proposed coder with entropy coder

Greyscale Image	Bit Rate	PSNR			SSIM		
		Wavelet	HWD	WBChh	Wavelet	HWD	WBChh
Elaine	0.0313	26.0787	26.0379	**26.1572**	0.7742	0.7736	**0.7772**
	0.0625	28.5023	28.3166	**28.5068**	0.8495	0.8461	**0.8543**
	0.125	**30.6054**	30.2521	30.519	0.9044	0.8988	**0.9084**
	0.25	**32.02**	31.6714	31.9928	0.9395	0.9328	**0.943**
	0.5	33.2136	32.8697	**33.6894**	0.9565	0.9454	**0.9599**
	1	35.1332	35.1656	**35.7934**	0.9724	0.9705	**0.9769**
Lenna	0.0313	24.9483	24.9151	**25.149**	0.7472	0.7445	**0.7621**
	0.0625	27.2717	27.2278	**27.4756**	0.8373	0.8322	**0.8479**
	0.125	**30.177**	29.787	30.1554	0.906	0.8954	**0.9129**
	0.25	**33.2174**	32.6094	33.0334	0.9488	0.939	**0.9521**
	0.5	**36.2463**	35.5261	36.1571	0.9743	0.9686	**0.9757**
	1	39.0228	38.4496	**39.388**	0.9873	0.9852	**0.9897**
Barbara	0.0313	22.3953	22.2475	**22.4764**	0.6683	0.6574	**0.6794**
	0.0625	**23.4394**	23.1937	23.4387	0.7404	0.717	**0.7482**
	0.125	24.7244	**25.1286**	24.7882	0.7995	0.8046	**0.8097**
	0.25	27.4438	**27.7538**	27.2451	0.8857	0.8849	**0.9011**
	0.5	**31.5082**	31.1816	31.2641	0.9506	0.9416	**0.9551**
	1	**36.593**	35.7801	36.2436	0.9828	0.9775	**0.9844**
Car	0.0313	22.3984	22.207	**22.4746**	0.6749	0.6436	**0.6956**
	0.0625	24.199	24.1835	**24.3042**	0.7555	0.7438	**0.7665**
	0.125	26.3195	26.0812	**26.4341**	0.843	0.833	**0.8513**
	0.25	28.9382	28.5615	**29.1113**	0.911	0.9006	**0.9229**
	0.5	**32.2237**	31.7555	32.1236	0.9564	0.951	**0.962**
	1	37.133	36.2774	**36.9989**	0.9837	0.9795	**0.9856**
Baboon	0.0313	19.8251	19.8213	**19.9111**	0.453	0.4341	**0.4784**
	0.0625	20.4307	20.4179	**20.5404**	0.5405	0.5452	**0.548**
	0.125	21.4004	21.3973	**21.4578**	0.6603	0.6595	**0.6949**
	0.25	**22.8348**	22.7674	22.8279	0.7695	0.7677	**0.7701**
	0.5	**25.0598**	24.889	24.8354	0.8706	0.8653	**0.8864**
	1	**28.6125**	28.26	28.208	0.9415	0.9367	**0.9527**
Bicycle	0.0313	16.3805	16.3569	**16.7095**	0.4745	0.4589	**0.5115**
	0.0625	17.4895	17.4637	**17.7532**	0.6059	0.5856	**0.6388**
	0.125	18.8766	18.6707	**19.2409**	0.7261	0.6986	**0.75**
	0.25	21.0118	20.5325	**21.4779**	0.8404	0.8002	**0.8633**
	0.5	24.0488	22.8446	**24.8807**	0.9155	0.8747	**0.9245**
	1	26.973	25.6051	**29.2508**	0.9592	0.9323	**0.9703**
Peppers	0.0313	24.015	23.9424	**24.2007**	0.7492	0.741	**0.7584**
	0.0625	26.6425	26.5217	**26.709**	0.8411	0.8236	**0.85**
	0.125	29.5566	29.0519	**29.7501**	0.9014	0.8877	**0.9118**
	0.25	32.3505	31.4356	**32.6541**	0.9412	0.9281	**0.9516**
	0.5	34.5408	33.6836	**35.0973**	0.9637	0.9549	**0.971**
	1	36.3182	35.629	**37.6365**	0.9764	0.9721	**0.9809**



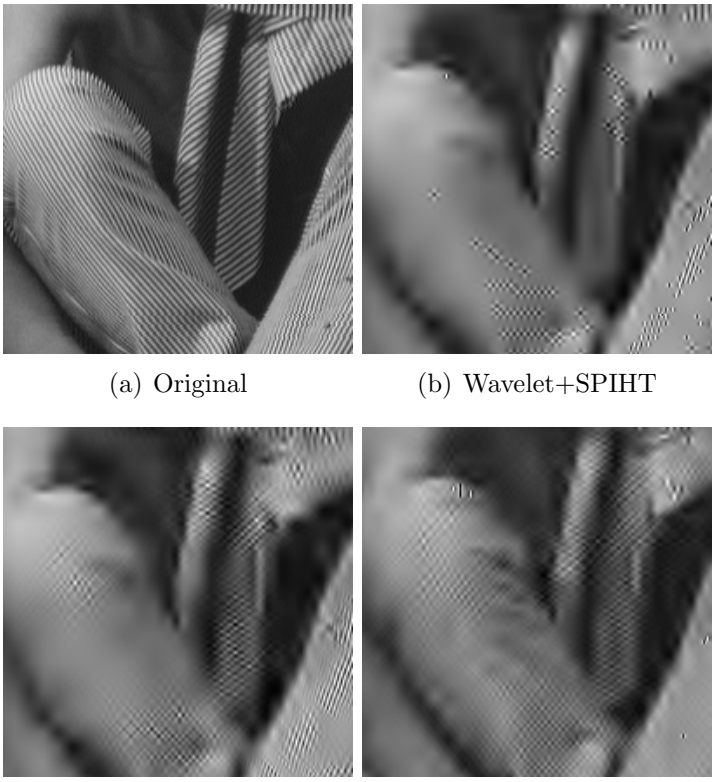
(a) Barbara Image (b) Lenna Image

(c) Bicycle Image (d) Peppers Image

Figure 5.13: PSNR comparison for greyscale image compression with entropy coding

close to that of the wavelet.

Performance also compared in terms of the PSNR and SSIM measures as shown in Table 5.4. It can be seen that the WBChh transform outperforms HWD and DWT at most rates, especially with images that have medium frequency content like Barbara and Bicycle. With the Lenna and Peppers images which have low frequency content, the SSIM performance of the WBChh transform are quite well compared to HWD, but very closely to the wavelets at certain bit rates. Overall, the WBChh shows the best performance in the SSIM measure owing to the fact that this coder is able to capture structural information efficiently. An additional entropy coding also continues to show similar performance in term of SSIM measure. The WBChh coder outperforms the HWD and wavelet SPIHT in



(a) Original (b) Wavelet+SPIHT

(c) HWD+SPIHT (d) WBChh

Figure 5.14: Barbara at bit rate 0.0625 (zoom)

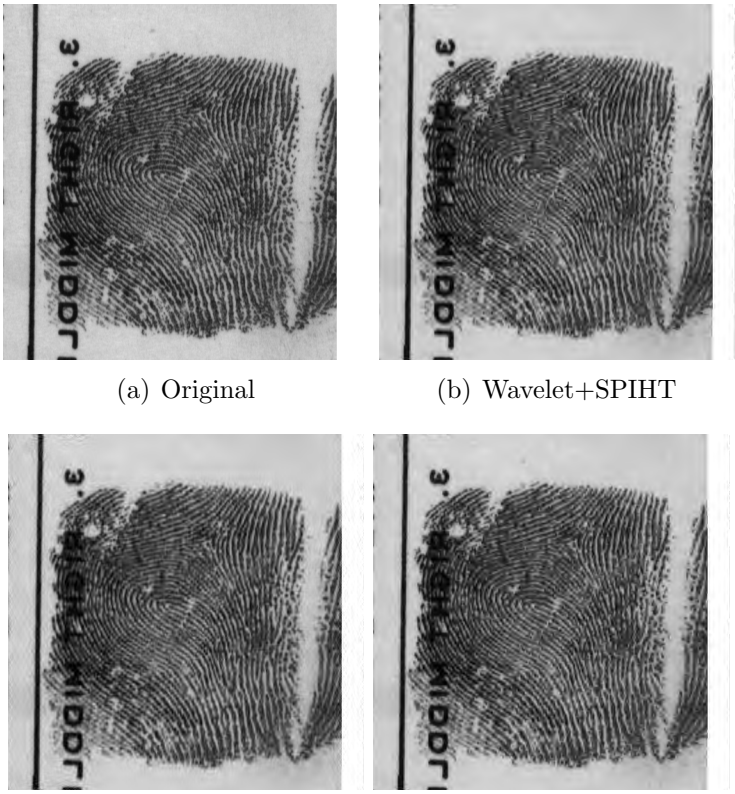
term of SSIM at all bit rates but only at certain rates in term of PSNR because it is a logarithmic measure. The results shown in Table 5.5 are the performance of the proposed coder with standard test images using entropy coding based on Equation 2.5.

5.4.3 Fingerprint application

Based on the results for image coding, the proposed transform coding can be deemed suitable for application in fingerprint compression schemes because of its ability to capture structural information. Fingerprint compression is important due to the increasing number of fingerprint records. Wavelet/scalar quantization (WSQ) was developed by the U.S. Federal Bureau of Investigation in 1993 to digitize its fingerprint database [96, 97]. WSQ is realized using a wavelet CDF-9/7 filter with scalar quantization and Huffman coding. The DFB for fingerprint matching and extraction is introduced in [87]. Zhang and Moloney [98, 94] introduced fingerprint compression based on the nonredundant contourlet transform (NRCT). However, although the result show significant improvements over wavelet, NRCT is high in complexity because the transform uses many types of complex filter to realize directionality in the wavelet domain to produce a transform with similar decompositions to that in the contourlet. Furthermore, since the transform decomposition is similar to that of the contourlet, the performance is only good at low bit rates (below 0.5 bit per pixel) as shown in the resulting values of PSNR [98, 94].

Implementation

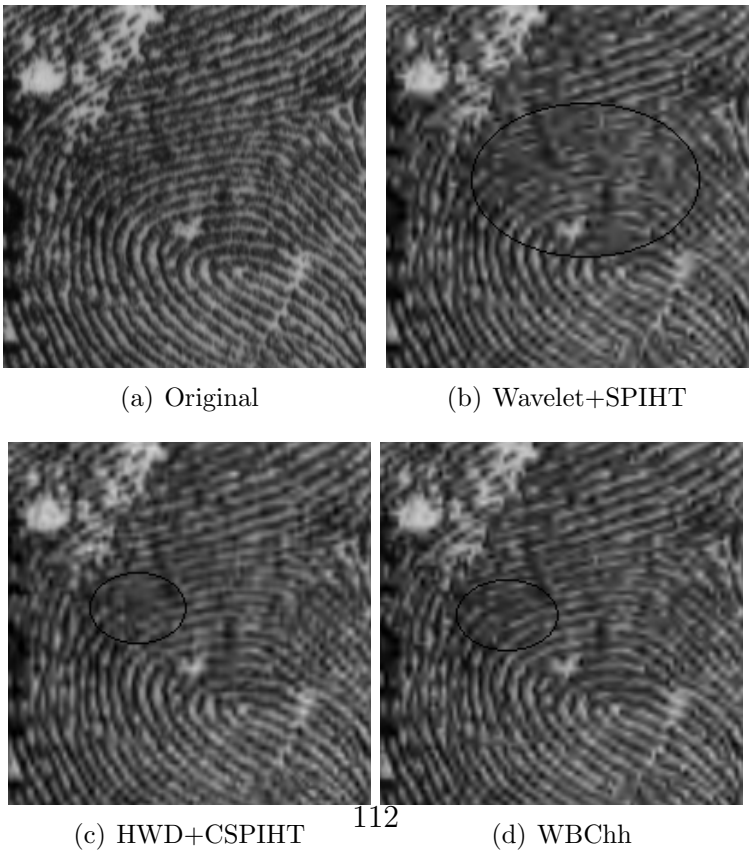
For the implementation test, a fingerprint database [99] was used. The aim of these test are to compare the transform based compression scheme among different transform with its quantization technique, so the entropy coding of all transform is simplified using Equation 2.5. In the future, this work can be extended to include proper entropy coding for performance comparison of fingerprint compression with FBI WSQ standard [96, 97]. As with the image coding technique, the proposed transform was coded with the listless specK to provide efficiency in terms of the memory required during coding and complexity. An example of a reconstructed fingerprint (f14) compressed at a bit rate of 0.25 is shown in Figure 5.15, with its zoom image for clear comparison in Figure 5.16.



(a) Original (b) Wavelet+SPIHT

(c) HWD+CSPIHT (d) WBChh

Figure 5.15: Fingerprint (f14) at bit rate 0.25



(a) Original (b) Wavelet+SPIHT

(c) HWD+CSPIHT (d) WBChh

Figure 5.16: Zoom fingerprint (f14) at bit rate 0.25

Table 5.6: Fingerprint compression

Greyscale Image	Bit Rate	PSNR			SSIM		
		Wavelet	HWD	WBChh	Wavelet	HWD	WBChh
f01	0.0313	**21.6273**	21.6055	21.5485	0.3793	0.3736	**0.4061**
	0.0625	23.1888	23.1136	**23.2569**	0.5703	0.5568	**0.5854**
	0.125	25.1595	25.0691	**25.3372**	0.7316	0.7283	**0.7845**
	0.25	**28.1007**	27.9785	28.0693	0.8864	0.8784	**0.8975**
	0.5	31.5962	31.4263	**32.0143**	0.9599	0.9551	**0.9676**
	1	**36.214**	35.952	36.1283	0.9888	0.9874	**0.9907**
f02	0.0313	**20.326**	20.3191	20.2859	0.3911	0.3746	**0.3993**
	0.0625	22.1744	22.0444	**22.6581**	0.651	0.6248	**0.7138**
	0.125	24.7855	24.7078	**25.1096**	0.8349	0.8252	**0.8588**
	0.25	27.9037	28.0315	28.0168	0.9236	0.9264	**0.9324**
	0.5	32.0123	31.8117	**32.397**	0.9777	0.9755	**0.9812**
	1	36.4054	36.1878	**36.7851**	0.994	0.993	**0.9946**
f03	0.0313	24.3193	24.2192	**24.4156**	0.5004	0.4944	**0.5057**
	0.0625	25.6458	25.5821	**25.7124**	0.6348	0.628	**0.669**
	0.125	27.6822	27.5968	**27.6952**	0.7976	0.7849	**0.8034**
	0.25	29.8841	29.8369	**30.0517**	0.8963	0.8972	**0.899**
	0.5	32.8167	32.6776	**32.9654**	0.9578	0.9563	**0.961**
	1	36.9505	36.6465	**37.0995**	0.9859	0.9846	**0.9872**
f09	0.0313	20.6209	**20.6728**	20.6329	0.4504	0.4479	**0.4718**
	0.0625	22.5132	22.2717	**22.9355**	0.6794	0.6374	**0.6957**
	0.125	24.838	24.8862	**28.0644**	0.8124	0.8084	**0.8415**
	0.25	27.9217	**28.0644**	27.9911	0.9151	0.9136	**0.9234**
	0.5	31.7024	31.5943	**32.1394**	0.9714	0.9686	**0.9769**
	1	36.1504	35.9633	**36.319**	0.9906	0.9899	**0.9924**
f14	0.0313	**21.0423**	21.0415	21.0184	0.5294	0.5254	**0.534**
	0.0625	22.6195	22.5472	**22.8736**	0.695	0.6705	**0.7007**
	0.125	**24.9654**	24.9276	24.845	0.8142	0.8106	**0.8274**
	0.25	28.4259	28.2991	**28.4544**	0.9152	0.9098	**0.9234**
	0.5	32.431	32.148	**32.6201**	0.9658	0.9587	**0.9689**
	1	**37.4964**	36.9313	37.3763	0.9842	0.9825	**0.9875**

The performance in term of PSNR and SSIM is shown in Table 5.6. Implementation to fingerprint images shows that the proposed WBChh coder consistently outperformed the wavelet and HWD CSPIHT in the SSIM measure. In terms of PSNR, a close performance between the wavelet and the WBChh can be seen, with the proposed transforms show better performance in most images. Despite this, the proposed WBChh coder have better performance than the wavelet and HWD because fingerprint images contain contour which indicate structural information in the image. This means that, for this comparison structural similarity is a more reliable measure compared to PSNR, a logarithmic measure. Moreover, the SSIM performance is consistent until up to bit rate 1 bits per pixel (bpp). The WBChh added only minimal complexity to wavelet transform but has the lower complexity compared to other directional-based transforms. This good performance indicates that the WBChh may be an attractive alternative to current fingerprint compression schemes.

5.5 Conclusions

This chapter describes the directionality in image transform. The previous hybrid wavelet and directional filter banks transform is improved with the introduction of wavelet-based contourlet in highpass domain that implement directional filter only to high frequency subband. The directional transform is restricted to highpass subbands in order to avoid the generated coefficients from frequency scrambling. This is due to the fact that frequency scrambling generates a noise like effect known as the pseudo-Gibbs phenomenon in the reconstructed compressed image. The directional transform used is based on the contourlet's directional filter. The proposed transform managed to provide a balanced performance which maintained the edges in textured images at low bit rates without introducing pseudo-Gibbs artefacts at higher bit rates with smooth images. The proposed transform is combined with a listless coding technique that uses fixed memory, so only minimum complexity was added compared to wavelet transforms while maintaining efficient encoding and decoding. The non-linear approximation comparison for the proposed transform is in between the wavelet and higher complexity directional-based transform such as contourlet and HWD-F. The per-

formance of the WBChh in image coding yielded good results in term of PSNR and SSIM with all images tested with or without entropy coding. An example of a fingerprint compression application showed the potential implementation of the WBChh with a specific image type. In the next chapter, the performance of the WBChh in colour image compression is investigated.

Chapter 6

Wavelet and Directional Filter Bank in Colour Image Compression

This chapter investigates the implementation of hybrid wavelets and the directional filter bank (DFB) for colour image compression. Hybrid wavelet and directional filter banks (HWD) apply the DFB in the wavelet domain to improve directional information of the wavelet transformed image. An example of an HWD is the wavelet-based contourlet transform (WBCT) [16] which was later introduced as the HWD-F [92, 17]. The HWD has been applied to greyscales and image produces better visual quality especially with images contain texture and contours. The WBCT is implemented using a DFB that has been developed for the contourlet transform. A performance comparison using non-linear approximation(NLA) shows that the WBCT performs very well compared to the wavelet and contourlet. However, for image coding WBCT only performs better at lower bit rates, due to artefact generated by the transform in the reconstructed compressed image. These artefacts are noise like features known as the pseudo-Gibbs phenomenon. In practice, image coding performance also depends on the choice of encoder and decoder used. Wavelets have considerable advantage because the zerotree and zeroblock coding techniques are able to efficiently capture the necessary information/coefficients. A balanced approach using the wavelet transform

matched with the efficient coding of colour image is expected to improve coding performance. However, despite good performance in greyscale compression, directional-based transforms have not attracted much attention from the research community for their ability to compress multispectral images, in this case colour images. Therefore, this chapter investigate the implementation of the DFB in wavelet domain for composite technique of colour image compression. Composite colour image compression is preferred to the separate compression of each colour band because it is able to offer the precise control of bit rates during progressive coding. The composite procedure also benefits from full embeddedness and the automatic allocation of bits among the colour planes. In addition, non-linear dependencies at high transition regions (such as edges) which remain among the spectral planes can be exploited during the compression process.

6.1 Motivation

The objective of this chapter is to investigate the directional filter bank in the wavelet domain for composite colour image coding. Given the improvements seen with greyscale images, introducing directional filters during the compression process is expected to improve the visual quality of reconstructed compressed colour images. In this case, composite progressive coding is used instead of the separate coding of each colour band. The composite technique will benefit from further exploitation of channel/spectral redundancy. Previous research on colour images that using directional transforms such as contourlets has been presented by Nabil and Peter [100]. However, this analysis was only based on approximations using a simple threshold for the transformed coefficients, and performance was measured based on entropy values and compression ratios. This only gives a hint of the potential performance in practical applications. The transform used for each colour component was a contourlet-wavelet denoted as CVT [100], which was then compared with normal wavelet decompositions. Moreover, the contourlet-wavelet transform used in that study was the same as previously proposed for greyscale images [15], which introduced a redundancy during coding. Despite giving good visual quality at lower bit rates, the redundancy that exists with contourlets, degrades coder performance during compression when compared to that of stan-

dard wavelet transforms. It is hoping that the present work will provide a clear view of practical capabilities of directional-based transforms for composite colour coding. Unlike in the previous implementation [100], the directional transforms here do not involve redundancy, since the directional filter is applied within the wavelet domain. In addition to its directionality, the selected transform adds only minimal complexity since it is implemented only in the luminance plane. A contourlet-based DFB implemented in the wavelet domain is predicted to be able to detect point of discontinuity produced by the tensor product in the wavelet. The implementation of a directional filter should increase the complexity of the wavelet transform. In order to minimize its complexity, directional information is applied only in certain subbands where it is expected to capture the point of discontinuities into fewer coefficients representing contours and edges. In theory, this should result in a better quality of compressed images since the most important information is contained in the luminance plane. However, this will be limited to certain bit rates only, since more significant coefficients are generated by the transform compared to the normal wavelet.

6.2 Colour image compression

To exploit spectral redundancy, the original images were transformed into decorrelated colour space. In this case, the standard colour transform for digital images, $YCbCr$ was used. The proposed transform has been tested with greyscale images in Chapter 5. For chrominance, Cb and Cr planes, the wavelet transform is applied, and hence this approach is a hybrid of a directional transform and wavelet. Implementation in the luminance plane aims to preserve the edge and directional information in the image. The flowchart for the implementation of this coding is described in Figure 6.1. An embedded coder based on a listless zeroblock structure was chosen instead of the state of the art zerotree, SPIHT and SPECK because it has lower complexity and performs efficiently with a listless structure. Listless coders use fixed memory, which reduces the memory required during coding. Location information represented by coordinates in lists that results in dynamic memory requirement in normal coder during the coding process are removed with the introduction of fixed scanning or mapping. This type of

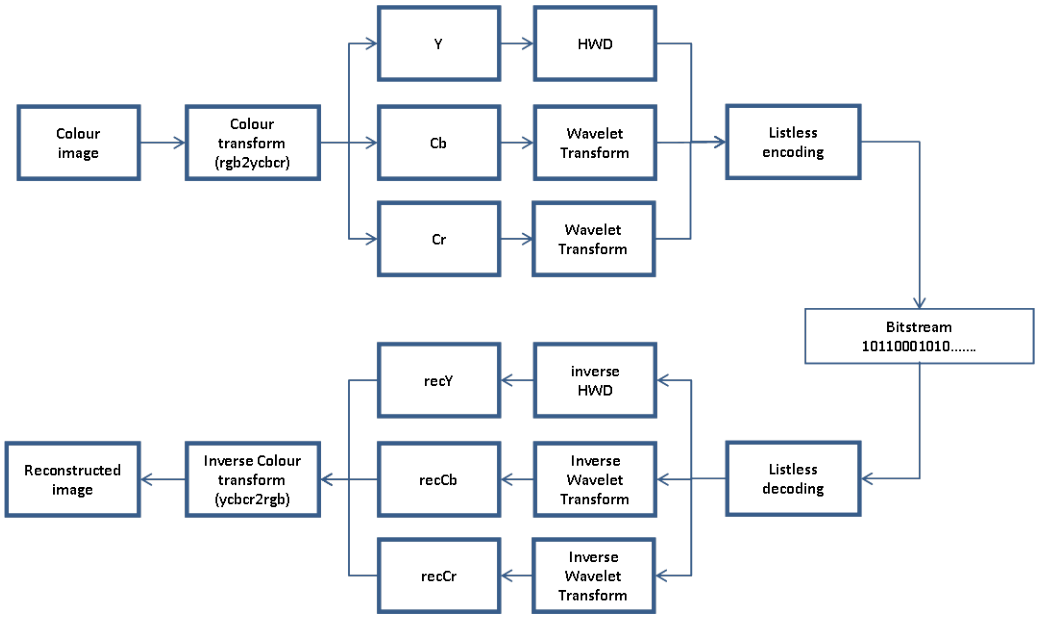


Figure 6.1: Proposed colour image coding

mapping is known as Morton ordering or Z-curve. It has space filling properties and preserves the locality of data points which match the wavelet subband decompositions. The Z-curve mapping is efficient in construct quadtrees which follow zeroblock partitioning rules. A block chart representing this implementation is shown in Figure 6.2.

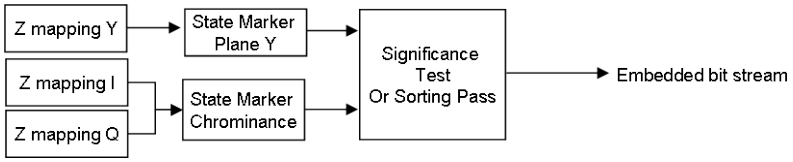


Figure 6.2: Encoding technique used for the proposed transform

6.2.1 Directional transforms in the wavelet domain

Firstly in this section, the proposed transform is reviewed. Figure 6.3 shows a schematic diagram of the proposed transforms to be applied in the luminance plane. The implementation of directional based transforms only to luminance plane is aimed to introduce directionality to composite colour coder without

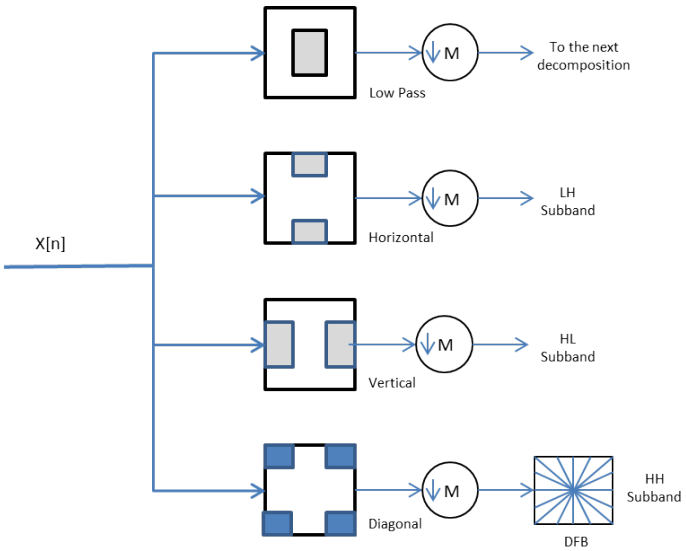
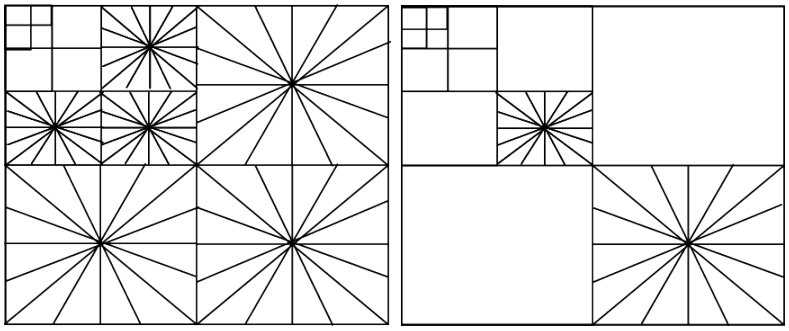


Figure 6.3: Schematic diagram of proposed HWD transform to luminance plane



(a) WBCT/HWD (b) WBChh

Figure 6.4: The schematic plot of the transforms with 4 levels of dyadic wavelet decomposition and 8 (2^3) directions at the two finest level

adding too much complexity to the coder. Figure 6.4 then shows an example of two HWDs where directional decomposition is overlaid on the wavelet decomposition. The reasons behind the implementation of both directional-based transforms are directional capabilities and non-redundant properties. Having directionality overlaid on the wavelet transform means that the coding process can be implemented using embedded bit plane coding, which is a type of scalar quantization that involves low complexity in the coding. Contradict to greyscale implementations, both transforms can be referred to as wavelet based contourlet (WBC) because the directional filter bank is from the contourlet transform. The reason for avoiding Laplacian pyramid filter used in the contourlet is redundancy factor of the filter, which is about 4/3. Redundancy is not suitable for compression application since more bit required to represent compressed images. Conversely, the HWD notation is used to refer to implementations that apply the directional filter to only a few fine levels [17] instead of all levels as used in some previous research [84, 18, 85]. From the schematic diagram, it is clear that the WBCT/HWD involve higher levels of computational complexity than the wavelet-based contourlet in the highpass domain (WBChh), because the directional filter is applied to each subband decompositions. The level of complexity of directional-based transforms have been discussed in Section 5.3.2. The main highlight of the implementation of the WBChh is to remove the pseudo-Gibbs artefacts generated by the WBCT due to frequency scrambling. At the same time, the WBChh is able to extract directional information from the image so that curves or singularities can be represented with fewer coefficients. Having said that, despite fewer coefficients for singular(edge), overlaid the DFB in a wavelet will generate more coefficients on top of the wavelet coefficients. So, as bit rate increased, the performance of proposed coefficients will be slightly lower, not because of any degradation in performance but because there are more significant bits generated compared to wavelet. This is because the wavelet transform on its own is a form of simple directional basis as described in Section 5.3.1. Directional filter bank applied in diagonal subband off wavelet decomposition will represent the singularities with fewer significant coefficients.

6.2.2 Embedded colour image coding

For colour image coding, a true colour image is transformed into decorrelated color space such as YIQ or YCbCr before a transform is applied to each plane. Composite coding benefits from precise control of bit rates and enable the exploitation of non-linear dependencies among spectral planes. Embedded colour image coding in the wavelet domain treats all colour planes as one unit at the coding stage before generating one mixed bit-streams. For most natural multispectral images in the YCbCr colour space, in addition to the redundancy of each component, the two chrominance components Cb and Cr are correlated in terms of insignificant coefficients within each chrominance plane compared with a given threshold. In this implementation, the listless colour set partitioned embedded block coder (SPECK), as introduced in Chapter 3, is used to exploit these properties. The listless coding is represented in Figure 6.2.

6.3 Numerical results and discussion

The following results were obtained with 24 bits colour image downloaded from USC-SIPI database [62] except for the cropped Barbara images [64]. The images were transformed into decorrelated colour space using YCbCr in Matlab, as described in Figure 6.1. Five level of decompositions of dyadic wavelet using the CDF-9/7 tap-filter applied on each colour plane. Directional decomposition on luminance plane in the $\{l_j\}_{5>j>1} = \{0, 0, 0, 3, 3\}$ where l_1 refers to the finest decompositions.

6.3.1 Performance measurement

In this work, peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) measures were used as performance indicators. Visual representation are provided to prove structural quality of compressed image. It must be stressed that the proposed method applied the new transforms to the luminance plane only and at the same times applied progressive coding to take advantage of the similarity between chrominance planes. Hence, for fair comparison, the same embedded

coding was applied to the transform tested and SPECK-based wavelet is set as a benchmark.

Rate distortion performance

The quality of the reconstructed images were evaluated in term of PSNR. For colour image of 24 bits per pixel (bpp), the overall PSNR RGB channel is defined as follows:

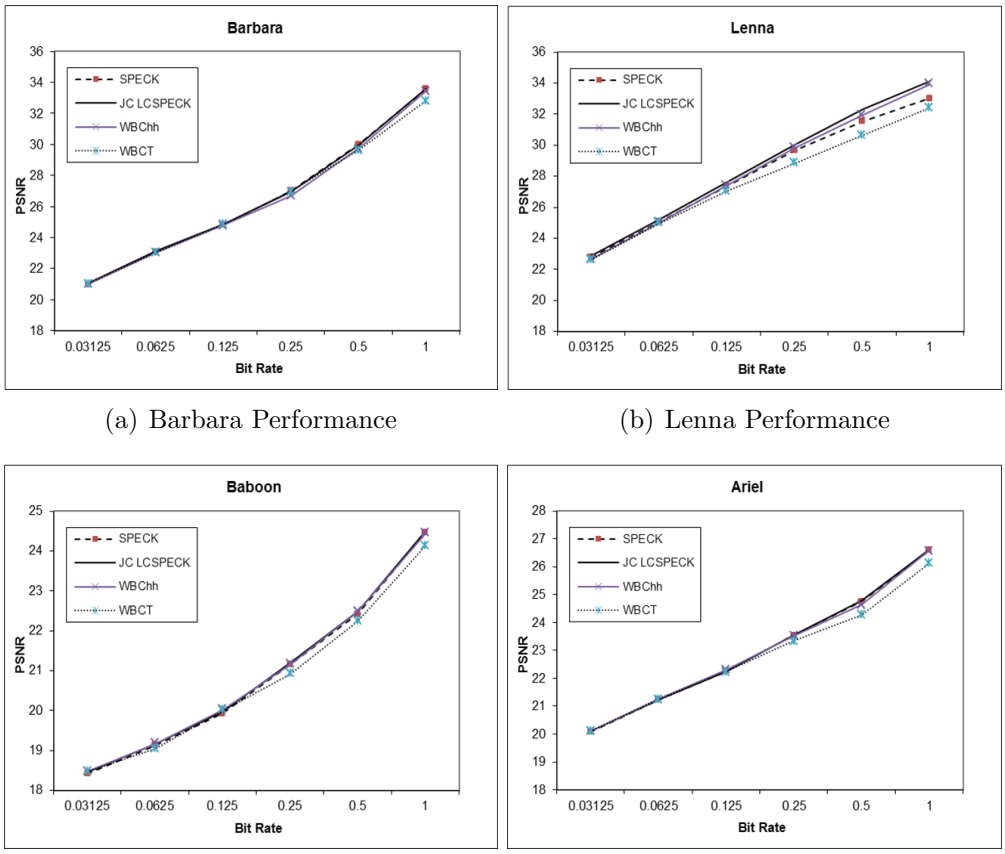
$$PSNR = 10log_{10}\frac{3 \times 255^2}{MSE_R + MSE_G + MSE_B}dB \tag{6.1}$$

with mean square error $MSE_{cp}R, G, B$ for each colour plane defined as

$$MSE_{cp} = \frac{1}{N}\sum_{i=0}^{n_1}\sum_{j=0}^{n_2}\left[x\left(i,j\right) - r\left(i,j\right)\right]^2 \tag{6.2}$$

where $x\left(i,j\right)$ and $r\left(i,j\right)$ are the original and reconstructed images respectively and $N = n_1 \times n_2$ is the number of pixels in the original image. The results based on the overall PSNR in the RGB channel are shown in Figure 6.5. From these results, the proposed implementation outperforms the wavelet SPECK in certain rates and perform very closely to wavelet with listless coding (ListlessJC in Table 6.1). At very low bit rates from 0.0313 to 0.125, the proposed colour coding performs very close to the WBCT transform, whereas at higher bit rates, the WBChh outperforms the WBCT in term of PSNR. A visual comparison of the reconstructed images at low bit rates is shown in Figure 6.6. Full results of some of the tested image is shown in Table 6.1.

From the results in Table 6.1, the performance of the WBChh transform with the listless coder is varies from image to image. The DFB applied to the wavelet transform add more significant information to the transform coefficients. If the image contains more directionality curves or singularities, it improve the compressed image. Actual comparison based on the compressed image visually proven the claim. The performance gain compare to wavelet with same coding is between ± 0.3 dB, which indicate a very close and tight performance. A comparison to wavelet with the original SPECK coding, show higher gain around 0.9 dB and decrease performance around ± 0.2 dB in some cases. A good PSNR is shown for



(a) Barbara Performance

(b) Lenna Performance

(c) Baboon Performance

(d) Ariel Performance

Figure 6.5: Overall PSNR in RGB over wavelet+SPECK coding

Table 6.1: Performance of the proposed colour coder

Bit Rate	PSNR				SSIM			
	SPECK	ListlessJC	WBChh	WBCT	SPECK	ListlessJC	WBChh	WBCT
				Lenna				
0.03125	22.7762	**22.8634**	22.649	22.608	0.6697	**0.6827**	0.6683	0.6662
0.0625	25.0783	**25.2456**	25.0349	24.9715	0.7918	**0.8084**	0.8014	0.7935
0.125	27.3691	**27.6238**	27.4048	27.0286	0.8742	**0.8905**	0.8805	0.8609
0.25	29.6657	**30.0306**	29.812	28.8302	0.9254	**0.9394**	0.9289	0.8989
0.5	31.5545	**32.3129**	31.9266	30.6418	0.9551	**0.9628**	0.9584	0.9416
1	33.0276	**34.1213**	33.9325	32.4016	0.9764	**0.9787**	0.9779	0.9709
				Barbara				
0.03125	20.9874	21.0362	21.0012	**21.046**	0.5903	**0.5936**	0.5907	0.5924
0.0625	23.0628	**23.1434**	23.0488	23.0481	0.7254	**0.7335**	0.7221	0.7176
0.125	24.8147	24.8647	24.7814	**24.865**	0.8144	0.8142	0.8131	**0.8226**
0.25	**26.998**	26.9673	26.7043	26.9877	0.8974	**0.9005**	0.8959	0.8934
0.5	**30.0118**	29.9876	29.7465	29.6297	0.951	**0.9548**	0.9529	0.943
1	33.5733	**33.6082**	33.4484	32.8327	0.9826	**0.9829**	0.9816	0.9768
				Baboon				
0.03125	18.4247	18.4536	**18.4773**	18.474	0.375	**0.3851**	0.3761	0.3756
0.0625	19.128	19.162	**19.1672**	19.0441	0.494	0.4925	0.4925	**0.5037**
0.125	19.9299	19.9836	**20.0192**	20.0073	0.6267	0.6522	**0.6567**	0.6516
0.25	21.159	**21.1866**	21.1596	20.9169	0.7335	**0.737**	0.7329	0.7265
0.5	22.4295	**22.486**	22.4844	22.2439	0.8558	**0.8701**	0.8685	0.8582
1	24.4759	**24.5034**	24.4422	24.1291	0.9187	**0.9418**	0.9404	0.9313
				Ariel				
0.03125	20.0742	20.0876	**20.1026**	**20.1026**	0.3652	0.3722	**0.3748**	**0.3748**
0.0625	21.2354	21.2406	**21.2596**	21.2585	0.5134	0.515	0.5191	**0.5208**
0.125	22.2481	22.25	**22.3116**	22.2342	0.6748	0.6897	**0.6923**	0.6886
0.25	23.542	**23.5451**	23.5314	23.3374	0.7795	**0.811**	0.8103	0.7977
0.5	24.763	**24.8007**	24.6476	24.2757	**0.8783**	0.8759	0.8742	0.8671
1	26.6029	**26.6096**	26.5857	26.1293	0.9453	**0.9518**	0.951	0.9438
				Ariel2				
0.03125	18.3924	18.4427	**18.4928**	18.4811	0.2949	0.3063	**0.3079**	0.3058
0.0625	19.0803	19.1234	**19.1978**	19.1716	0.4725	0.4869	**0.4928**	0.4902
0.125	19.9849	20.1107	**20.1493**	20.097	0.611	0.6495	**0.6501**	0.645
0.25	21.0682	**21.1009**	21.0925	20.885	**0.765**	0.7631	0.7609	0.7475
0.5	22.5344	**22.7294**	22.6823	22.4739	0.8678	**0.8909**	0.8867	0.8746
1	24.8855	**24.9614**	24.863	24.4063	0.9383	**0.9503**	0.9459	0.9327
				Plane				
0.03125	21.6678	21.6253	**21.7128**	21.7025	**0.7009**	0.6987	0.6894	0.6882
0.0625	23.7688	23.7875	23.8441	**23.8608**	0.8004	**0.8074**	0.7956	0.7671
0.125	25.9797	**25.9895**	25.8356	25.6922	0.8663	**0.8669**	0.8618	0.841
0.25	28.1736	**28.311**	28.1393	27.4254	**0.9257**	0.9251	0.9224	0.9004
0.5	31.4892	**31.5224**	31.3447	30.5488	**0.9607**	0.9601	0.9581	0.943
1	34.9223	**35.1019**	34.8831	33.7	**0.9801**	0.9793	0.9784	0.972

high frequency images such as Ariel and Baboon at low bit rates. A comparison between the WBChh and WBCT transforms shows that WBChh outperform WBCT in most rates, except at certain very low bit rates around 0.2 dB.

Structural Similarity

Structural similarity (SSIM) [25] is a reference based image quality measure that computes using the combination of nonstructural and structural distortion. The equation for this method is presented in Equation 2.9. For colour images the measurement is applied to the luminance plane where the image is transformed to a greyscale image using the 'rgb2gray' function in Matlab before SSIM is measured [25]. An interesting feature of using the SSIM metric is its sensitivity to three common types of distortions: loss of correlation, luminance distortion and loss of contrast [101]. The SSIM results in Table 6.1 show the inconsistent performance of the proposed transform from image to image, as with PSNR. It seems that, at most bit rates, the wavelet with listless coding has better SSIM. However, one of the reasons that contribute to this is that SSIM is measured using the greyscale value of compressed colour image. SSIM is a reliable measure in the analysis if colour image especially when comparing similar transforms. Previous research on colour image measurement [101], has highlighted the fact that most of the performance measures were based on metrics developed for use with the greyscale images. This is because colour images has more properties than just luminance and brightness. Examples of compressed images at low bit rates are presented for reader observation.

The original figure is cropped to zoom the area that contain structural information such as curves and texture. From Figure 6.6, it can be seen that at a bit rate of 0.25, the best visual quality is shown by proposed coder (WBChh), despite the fact that, the best overall PSNR in RGB domain is wavelet with SPECK coding at 26.998 dB. As shown in Figure 6.6(e), the texture in Barbara's scarf is preserved, but pseudo-Gibbs artefacts are evident in the smooth area. In Figure 6.6(c), all of the texture in the scarf is lost; and in Figure 6.6(g) the best results are achieved, maintaining the texture and and the quality of the smooth area. The HWD/WBCT coding provides lower performance compared to the

(a) Wavelet+ListlessJC at 0.125bpp

(b) Wavelet+ListlessJC at 0.25bpp

(c) Wavelet SPECK 0.125bpp

(d) Wavelet SPECK at 0.25bpp

(e) HWD at 0.125bpp

(f) HWD at 0.25bpp

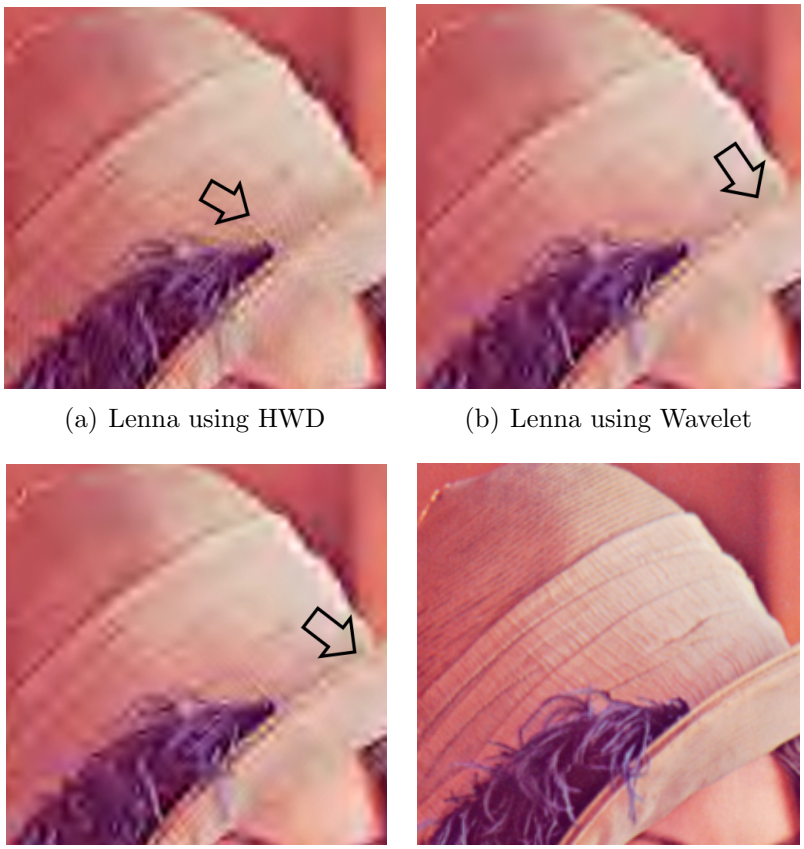
(g) WBChh at 0.125bpp

(h) WBChh at 0.25bpp

Figure 6.6: Comparison of part of Barbara image at low bit rates

wavelet in terms of distortion rate curves due to the pseudo-Gibbs artefacts. The artefact generated in the smooth region can be seen in Figure 6.6(f), in which the distortion rate lower compared to the wavelet coding. The Barbara image mostly consists of textures and oscillatory patterns, so this information is preserved at lower bit rates using the proposed transform. This results in superior visual quality at very low bit rates, especially with images that contain texture and fine details.

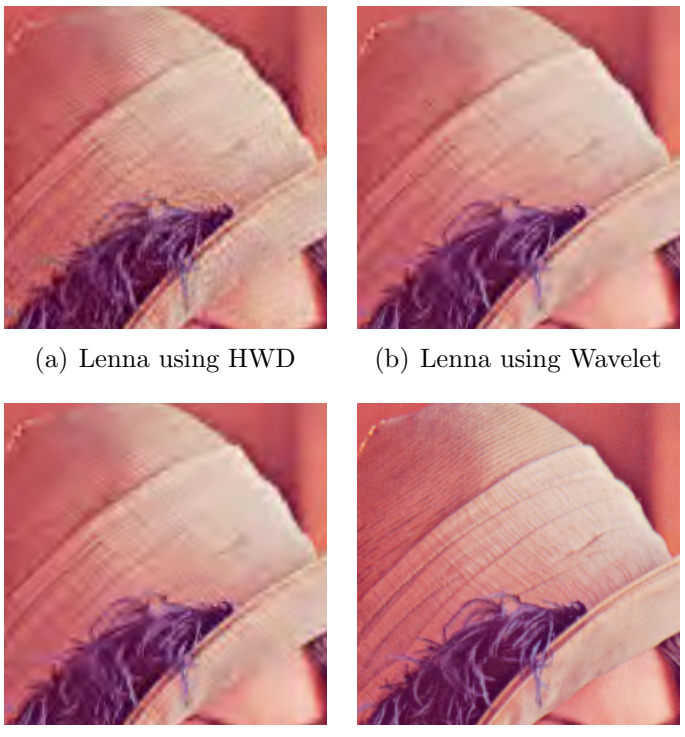
For the Lenna image as shown in Figure 6.7 at 0.125 bit rate, a hat line which is shown by the arrow clear in the WBChh compressed image compared to the wavelet. Another example at bit rate 0.25 is shown in Figure 6.8. Although the best values of PSNR and SSIM at bit rate of 0.125 and 0.25 are achieved using the wavelet, visually the directional-based transform preserves edges and curves better than the wavelet. This is true with all the tested images, despite lower overall measures in terms of PSNR and greyscale SSIM. Examples of compressed image for observation in visual quality comparison shown in Figures 6.9, 6.10 and 6.11.



(a) Lenna using HWD (b) Lenna using Wavelet

(c) Lenna WBChh (d) Original

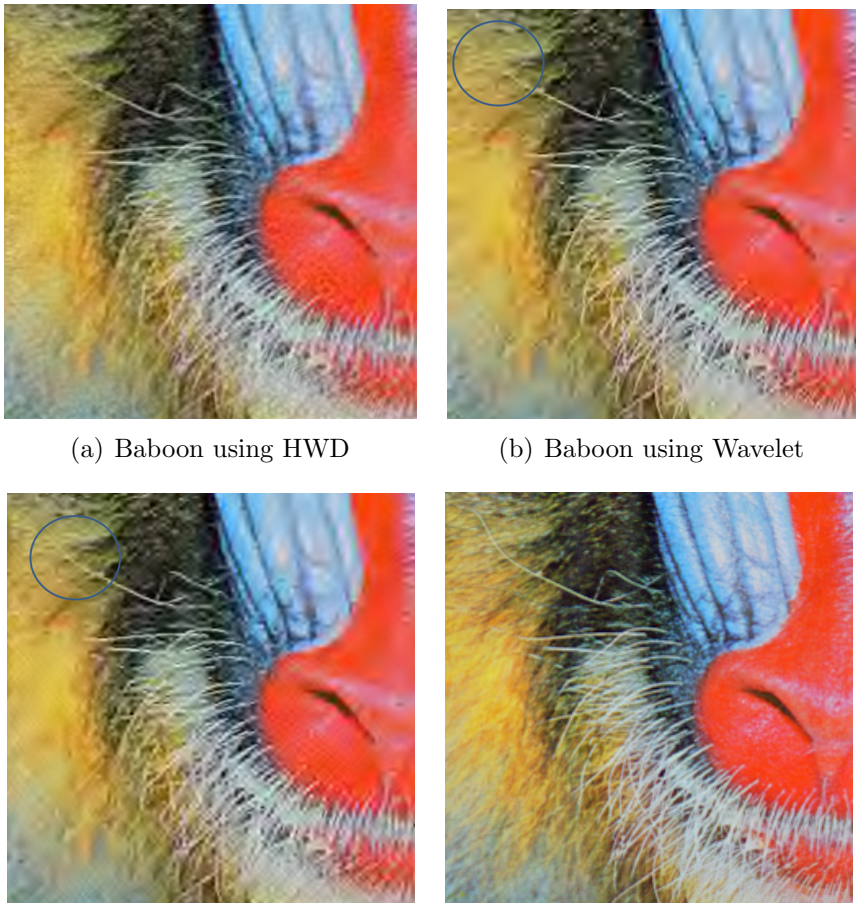
Figure 6.7: Comparison of part of Lenna image at 0.125 bit rates



(a) Lenna using HWD (b) Lenna using Wavelet

(c) Lenna WBChh (d) Original

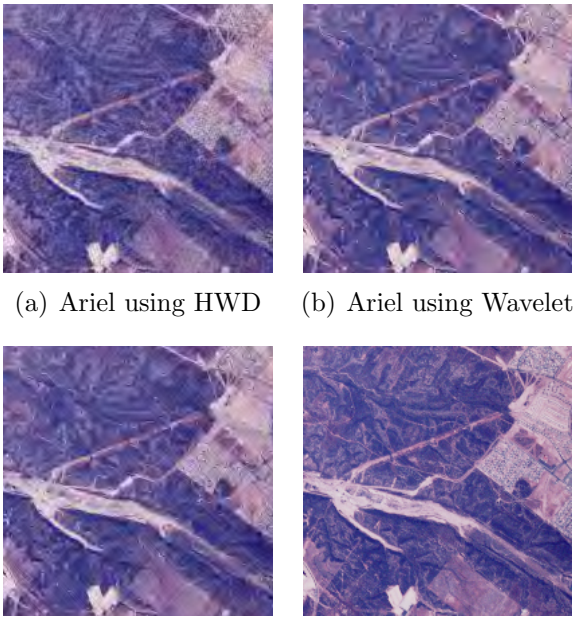
Figure 6.8: Comparison of part of Lenna image at 0.25 bit rates



(a) Baboon using HWD (b) Baboon using Wavelet

(c) Baboon WBChh (d) Original

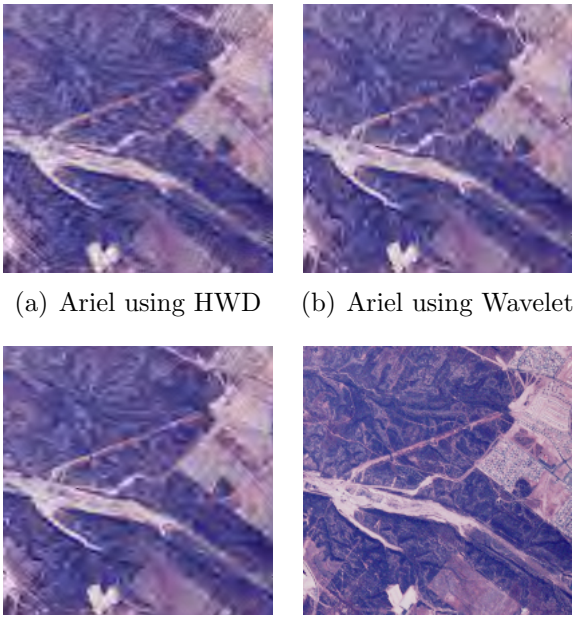
Figure 6.9: Comparison of part of Baboon image at 0.5 bit rates



(a) Ariel using HWD (b) Ariel using Wavelet

(c) Ariel WBChh (d) Original

Figure 6.10: Comparison of part of Ariel image at 0.5 bit rates



(a) Ariel using HWD (b) Ariel using Wavelet

(c) Ariel WBChh (d) Original

Figure 6.11: Comparison of part of Ariel image at 0.25 bit rates

6.3.2 Memory requirement and complexity analysis

In this chapter, the proposed coding technique uses directional filter bank overlaid over wavelet decompositions only to high-pass subband. So, the computational complexity as compared to other directional based transform is lower as discussed in Section 5.3.2. In term of colour image implementation, the directional based transform is only applied to luminance plane so a minimal additional complexity is added when compared to normal wavelet based compression. In short, the proposed transform complexity for colour image compression is in the middle between wavelet and WBCT. In term of memory requirement during coding, a comparison between listless algorithm and original SPECK has been discussed in Section 3.5.2. Overall, the proposed transform added minimal complexity to offer better directionality in compressed image while maintaining efficient coding process using listless structure that uses fixed memory with joined significant test on chrominance plane.

6.4 Conclusions

Progressive colour image coding using the contourlet-based directional filter bank (DFB), now termed the wavelet-based contourlet in the highpass domain (WBChh), has been developed. The transform has been coded with fixed memory and employs listless SPECK-based coding with joint significance tests for the chrominance planes. The proposed coding technique managed to outperform wavelet based coding in terms of distortion rate performance only at certain bit rates with certain types of image. SSIM was measured using the greyscale version of the compressed colour image, showing significant improvements to very limited rate. The WBChh managed to outperform the WBCT/HWD at most bit rates despite having lower complexity. The directional transform was implemented only in the luminance plane, owing to the fact that luminance contains the most important information for the human visual system. The implementation of directional filter bank to wavelet decomposition increased the numbers of significants, coefficients which consequently reduced performance as bit rate increased compared to the wavelet. However, it is clear that the additional of directional informa-

tion to the wavelet transform does significantly improve the visual quality of the reconstructed compressed image at lower bit rates, at least based on the visual quality of the reconstructed image. In conclusion, the main highlight of colour image coding using a directional based transform is not the reduction in the error of compressed image. Instead, the main contribution is the correct preservation and reconstruction of important features of an image, such as edges and texture even at very low bit rates. This would benefit application such as content-based retrieval in portable or mobile devices which have limited memory and lower bit rates capability.

Chapter 7

Conclusions

7.1 Contribution of the thesis

This thesis describes investigations into wavelet-based and scalable image compression. The following elements of the research are believed to be original contributions made in this study.

- Embedded colour image coding with listless set partitioned embedded block coder: the listless technique is very good at reducing the memory needed during the encoding process. Based on SPECK partitioning, the complexity of the partitioning algorithm is reduced. The scanning order is modified to enable exploitation of chrominance so that larger areas of the chrominance plane are tested, aiming to provide comparable performance for lossy compression and reduce the number of bit required for lossless compression. The algorithm performed quite well compared to JPEG2000, set partitioning in hierarchical trees (SPIHT) and the original CSPECK with lower working memory and complexity.

- Three dimensional listless SPECK: this is an extension of the two dimensional listless SPECK to three dimensions so as to be suitable for three dimensional (3D) source images. This implementation enabled the exploitation of redundancy in the spectral domain. Using a listless structure, the working memory required during coding was reduced compared to the normal 3D SPECK.

- The introduction of wavelet-based contourlet in highpass domain avoided the pseudo-Gibbs effect that arises from frequency scrambling in wavelet transforms. This enabled the transform to maintain smooth areas in the compressed image. The key to this contribution is the implementation of the directional filter bank only to highpass subband (HH) which contain no frequency scrambling in the wavelet domain. The proposed transform involves lower level of complexity than the previous hybrid wavelet and directional filter banks and the contourlets and providing non-redundant solutions appropriate for image compression application.

- The implementation of listless coding for colour image using WBCT consists of HWD and WBChh which was expected to improve colour image compression. This work builds on efficient encoding algorithm proposed, combining it with non-redundant transforms that offer directional information. The implementation of the proposed transform was applied to the luminance plane only in order to reduce the complexity associated with the directional filter. Further coding based on listless structures which combine scanning in the chrominance plane aimed to further improve the results. Although, little improvement was noted in term of PSNR and SSIM, visual inspection and observation show significant improvements in edge singularity and texture due to implementation of directionality in the transform.

7.2 Future Work

This thesis has investigated listless rate scalable coding and directional filtering in wavelet transform. However, there still remains scope for future research to further improve the performance of wavelet-based embedded coders, including in the area described below:

- The listless coders in this work were not implemented with entropy coding. Suitable entropy coding could improve the performance of proposed coder.

- The 3D listless coder implemented in this work is based on the 3D zeroblock coder. Further investigation based on the 2D SPIHT for 3D using a scan-

based listless method might further improve the outcome, given that the benefits of listless coding include reduced memory requirements.

- The directional filter bank implemented in this work is based on DFB filters developed in contourlets. Although they perform quite well compared with other type of filters, their filter lengths could be shorter, which could open a possible area of research for better performance.

- In evaluating colour image compression, most type of image measurement are based on error or use techniques which are available for greyscale images. Greyscale measures sometimes do not correlated with human visual systems especially when dealing with colour. Further research on the assessment of quality in colour images are needed.

References

[1] Khelifi, F. (2007) *Image Compression and Watermarking in the Wavelet Transform Domain*. Ph.D. thesis, School of Electronics, Electrical Engineering, and Computer Science, Queen's University of Belfast. 3

[2] Islam, A. (1999) *Set-Partitioned Image Coding*. Ph.D. thesis, Rensselaer Polytechnic Institute, United States – New York. 3, 4

[3] Shapiro, J. (1993) Embedded image coding using zerotrees of wavelet coefficients. *Signal Processing, IEEE Transactions on*, **41**, 3445 –3462. 5, 28, 31, 37

[4] Rao, Y. P. C., K. R. (2001) *The transform and data compression handbook*. Electrical Engineering and Applied Signal Processing Series, CRC Press. 5, 18, 21, 22

[5] Said, A. and Pearlman, W. (1996) A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *Circuits and Systems for Video Technology, IEEE Transactions on*, **6**, 243 –250. 5, 29, 30, 36, 37, 41

[6] Taubman, D. (2000) High performance scalable image compression with ebcot. *Image Processing, IEEE Transactions on*, **9**, 1158 –1170. 5, 33, 34

[7] Islam, A. and Pearlman, W. A. (1998) Embedded and efficient low-complexity hierarchical image coder. vol. 3653, pp. 294–305, SPIE, http://link.aip.org/link/?PSI/3653/294/1. 5, 36, 38, 43, 56, 66

[8] Mallat, S. (2008) *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 3rd edn. 6, 22, 23, 33

[9] Vetterli, M. (2001) Wavelets, approximation, and compression. *Signal Processing Magazine, IEEE*, **18**, 59 –73. 6, 86

[10] Cands, E. and Donoho, D. (1999) Curveletsa surprisingly effective nonadaptive representation for objects with edges. *Curves and Surface Fittingin : Saint-Malo*. 6, 86, 87

[11] Ma, J. and Plonka, G. (2010) The curvelet transform. *Signal Processing Magazine, IEEE*, **27**, 118 –133. 6, 86, 87

[12] Le Pennec, E. and Mallat, S. (2000) Image compression with geometrical wavelets. *Image Processing, 2000. Proceedings. 2000 International Conference on*. 6, 86, 88

[13] Peyre, G. and Mallat, S. (2005) Discrete bandelets with geometric orthogonal filters. *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 1, pp. I – 65–8. 6, 86

[14] Mallat, S. and Peyre, G. (2007) A review of bandlet methods for geometrical image representation. *Numerical Algorithms*, **44**, 205–234, 10.1007/s11075-007-9092-4. 6, 87

[15] Do, M. and Vetterli, M. (2005) The contourlet transform: an efficient directional multiresolution image representation. *Image Processing, IEEE Transactions on*, **14**, 2091 –2106. 6, 86, 88, 89, 90, 91, 93, 97, 99, 100, 102, 103, 117

[16] Eslami, R. and Radha, H. (2004) Wavelet-based contourlet transform and its application to image coding. *Image Processing, 2004. ICIP '04. 2004 International Conference on*, vol. 5, pp. 3189 – 3192 Vol. 5. 6, 89, 90, 106, 116

[17] Eslami, R. and Radha, H. (2007) A new family of nonredundant transforms using hybrid wavelets and directional filter banks. *Image Processing, IEEE Transactions on*, **16**, 1152 –1167. 6, 89, 90, 91, 96, 100, 106, 116, 121

[18] Zhang, S., Xue, X., and Shi, J. (2010) Image compression based on contourlet and no lists spiht. *Computer, Mechatronics, Control and Electronic Engineering (CMCE), 2010 International Conference on*, vol. 6, pp. 211 –214. 6, 89, 90, 121

[19] Parrilli, S., Verdoliva, L., and Poggi, G. (2008) A spiht-like image coder based on the contourlet transform. *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pp. 2908 –2911. 6

[20] Furht, B. (1998) *Image Presentation and Compression*, chap. 8. CRC Press, Inc., 1st edn. 10, 11

[21] Richardson, I. E. (2002) *Video Codec Design: Developing Image and Video Compression Systems*. John Wiley & Sons, Inc. 10, 16

[22] Peyre, G. (2010), Advanced signal and image processing. Course notes. 13, 14, 92

[23] Woods, J. W. (2006) *Multidimensional Signal, Image, and Video Processing and Coding*. Academic Press, Inc. 14, 25

[24] Salomon, D. (2004) *Data Compression: The Complete Reference*. third edn. 14, 26, 33

[25] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. (2004) Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, **13**, 600 –612. 17, 56, 100, 126

[26] Zhou Wang, H. R. S., Alan C. Bovik and Simoncelli, E. P. (2011), The ssim index for image quality assessment. Website, https://ece.uwaterloo.ca/~z70wang/research/ssim/ssim_index.m, February 2011. 18

[27] Ahmed, N., Natarajan, T., and Rao, K. (1974) Discrete cosine transfom. *Computers, IEEE Transactions on*, **C-23**, 90 – 93. 18

[28] Ghanbari, M. (2003) *Standard Codecs: Image Compression to Advanced Video Coding*. Institution of Electrical Engineers. 19, 29, 40

[29] Pennebaker, W. B. and Mitchell, J. L. (1993) *Jpeg: Still Image Data Compression Standard*. Van Nostrand Reinhold. 19

[30] Graps, A. (1995) An introduction to wavelets. *IEEE Comput. Sci. Eng.*, **2**, 50–61, http://portal.acm.org/citation.cfm?id=615244.615342. 20, 23

[31] Antonini, M., Barlaud, M., Mathieu, P., and Daubechies, I. (1992) Image coding using wavelet transform. *Image Processing, IEEE Transactions on*, **1**, 205 –220. 24, 25, 52

[32] Wasilewski, F. (2011), Wavelet properties. Website, http://wavelets.pybytes.com/wavelet/bior4.4/, 1 January 2011. 25

[33] Usevitch, B. (2001) A tutorial on modern lossy wavelet image compression: foundations of jpeg 2000. *Signal Processing Magazine, IEEE*, **18**, 22 –35. 25

[34] Peyre, G. (2010) The numerical tours of signal processing - advanced computational signal and image processing. Tech. rep., http://hal.archives-ouvertes.fr/hal-00519521/, June 2010. 26

[35] Skodras, A., Christopoulos, C., and Ebrahimi, T. (2001) The jpeg 2000 still image compression standard. *Signal Processing Magazine, IEEE*, **18**, 36 –58. 28, 85

[36] Pearlman, W. A. and Said, A. (2008) *Foundations and Trends in Signal Processing*, vol. Vol. 2, pp. 95–180. Now Publishers, 2 edn. 28, 38

[37] Pearlman, W., Islam, A., Nagaraj, N., and Said, A. (2004) Efficient, low-complexity image coding with a set-partitioning embedded block coder. *Circuits and Systems for Video Technology, IEEE Transactions on*, **14**, 1219 – 1235. 32, 38, 43, 54, 66

[38] Taubman, D. D. (2010), Kakadu software. Website, http://www.kakadusoftware.com/,January 2010. 33, 53

[39] Committee, J. (2007), Final committee draft for jpeg2000. Website, http://www.jpeg.org/jpeg2000/CDs15444.html,Sept 2007. 34

[40] Bayazit, U. (2011) Adaptive spectral transform for wavelet-based color image compression. *Circuits and Systems for Video Technology, IEEE Transactions on*, **21**, 983 –992. 36, 42

[41] Wheeler, F. and Pearlman, W. (2000) Spiht image compression without lists. *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on*. 36, 38, 42, 45, 46, 53, 67, 71

[42] Latte, M. V., Ayachit, N. H., and Deshpande, D. (2006) Reduced memory listless speck image compression. *Digital Signal Processing*, **16**, 817 – 824. 36, 38, 42, 44, 45, 53, 54, 62, 67, 71, 106

[43] Lin, W.-K. and Burgess, N. (1998) Listless zerotree coding for color images, nov. vol. 1, pp. 231 –235 vol.1. 38, 42

[44] Lin, W.-K., Ng, B. W.-H., Burgess, N., and Bouzerdoum, A. (1999) Reduced memory zerotree coding algorithm for hardware implementation, jul. vol. 2, pp. 57 –61 vol.2. 38, 42, 67

[45] Lin, W.-K., Moini, A., and Burgess, N. (2001) Listless zerotree coding using raster tree search. *Electrical and Electronic Technology, 2001. TENCON. Proceedings of IEEE Region 10 International Conference on*. 38, 42

[46] Lin, W.-K., Moini, A., and Burgess, N. (2001) Tree-pruning listless zerotree coding. *Multimedia and Expo, 2001. ICME 2001. IEEE International Conference on*, pp. 415 – 418. 38, 42

[47] M. Saenz, K. S., P. Salama and Delp, E. J. (1999) An evaluation of color embedded wavelet image compression techniques. *Proceedings of the SPIE Conference on Visual Communications and Image Processing*, San Jose, California, January 23-29, vol. 3653, pp. 282–293. 41

[48] Kim, B.-J. and Pearlman, W. A. (1998) Fast color-embedded video coding with spiht. *Proc. 1998 Data Compression Conference*, March. 41

[49] Shen, K. and Delp, E. (1997) Color image compression using an embedded rate scalable approach. *Image Processing, 1997. Proceedings., International Conference on*, Oct., vol. 3, pp. 34 –37 vol.3. 41

[50] Kassim, A. and Lee, W. S. (2003) Embedded color image coding using SPIHT with partially linked spatial orientation trees. *Circuits and Systems for Video Technology, IEEE Transactions on*, **13**, 203 – 206. 41

[51] Bouridane, A., Khelifi, F., Amira, A., Kurugollu, F., and Boussakta, S. (2004) A very low bit-rate embedded color image coding with SPIHT. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, May. vol. 3, pp. 689–692. 41, 42

[52] Khelifi, N. D. e. a., F. (2004) Compression of the color images by spiht technique. *International Conference on Information and Communication Technologies: From Theory to Applications, 2004. Proceedings*. 41

[53] Khelifi, F., Bouridane, A., and Kurugollu, F. (2008) Joined spectral trees for scalable spiht-based multispectral image compression. *Multimedia, IEEE Transactions on*, **10**, 316 –329. 41

[54] Brahimi, T., Melit, A., and Khelifi, F. (2009) An improved spiht algorithm for lossless image coding. *Digital Signal Processing*, **19**, 220 – 228. 41

[55] F. Khelifi, F. K., A. Bouridane (2005) Efficient lossless colour image coding with speck. *13th European Signal Processing Conference Eusipco 2005*, Antalya - Turkey, 4-8 September 2005. 42

[56] Khelifi, F., Kurugollu, F., and Bouridane, A. (2008) Speck-based lossless multispectral image coding. *Signal Processing Letters, IEEE*, **15**, 69 –72. 42

[57] Pan, H., Siu, W.-C., and Law, N.-F. (2008) A fast and low memory image coding algorithm based on lifting wavelet transform and modified spiht. *Signal Processing: Image Communication*, **23**, 146 – 161. 42, 67

[58] Li, W., Cui, Z., and Gao, L. (2010) Low bit rate image coding based on wavelet transform and color correlative coding. *Computer Design and Applications (ICCDA), 2010 International Conference on*, vol. 1, pp. V1–479 –V1–482. 42

[59] Shively, R., Ammicht, E., and Davis, P. (2000) Generalizing spiht: a family of efficient image compression algorithms. vol. 6, pp. 2059 –2062 vol.4. 42

[60] Lian, J., Wang, K., and Yang, J. (2006) Listless zerotree image compression algorithm. *Signal Processing, 2006 8th International Conference on*, 16. 42, 67

[61] Seetharaman, G., Zavidovique, B., and Shivayogimath, S. (1998) Z-trees: adaptive pyramid-algorithms for image segmentation. *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, Oct., pp. 294 –298 vol.3. 45, 46

[62] of Southern California Viterbi School of Engineering, U. (2010), The usc-sipi image database. Website, http://sipi.usc.edu/database/database.php?volume=misc, 20 January 2010. 52, 122

[63] of Granada, C. V. G. U. (2002), Test images. Website, http://decsai.ugr.es/cvg/dbimagenes/c256.php, October 2008. 52

[64] Vestola, M. (2007), Image compression. Website, http://www.mvnet.fi/kuvat/tutkielmat/image_comp/uncompressed/barbara.png, 15 January 2009. 52, 122

[65] Tang, X., Pearlman, W. A., and Modestino, J. W. (2003) Hyperspectral image compression using three-dimensional wavelet coding. vol. 5022, pp. 1037–1047, SPIE, http://link.aip.org/link/?PSI/5022/1037/1. 66, 76

[66] Tang, X. and Pearlman, W. A. (2005) *Hyperspectral Data Compression*, chap. Three-Dimensional Wavelet-Based Compression of Hyperspectral Images. Kluwer Academic Publishers. 66

[67] Tang, X., Cho, S., and Pearlman, W. (2003) Comparison of 3d set partitioning methods in hyperspectral image compression featuring an improved 3d-spiht, mar. p. 449. 66

[68] Tang, X. and Pearlman, W. (2004) Lossy-to-lossless block-based compression of hyperspectral volumetric data, oct. vol. 5, pp. 3283 – 3286 Vol. 5. 66

[69] Jing, L., Linhui, L., Teng, M., and Lidan, C. (2009) Improved listless zerotree image compression algorithm and application. *Image and Signal Processing, 2009. CISP '09. 2nd International Congress on*, pp. 1 –4. 67

[70] Dickau, R. (2008), Image of 3d morton scan. http://en.wikipedia.org/wiki/File:Lebesgue-3d-step2.png,Access on January 2010. 68

[71] University, A. S. (2008), Yuv video sequences. Website, http://trace.eas.asu.edu/yuv/,1 October 2008. 73

[72] (2009), Xiph.org test media. Website, http://media.xiph.org/video/derf/, October 2009. 73

[73] Mri images from auckland mri research group. Website, http://atlas.scmr.org/download.html,October 2010. 73, 80

[74] Peyre, G. (2009), Volumetric wavelet data processing. Website, http://www.ceremade.dauphine.fr/~peyre/numerical-tour/tours/multidim_2_volumetric/, October 2009. 73

[75] Lundeen, S. (2009), Airborne visible infrared imaging spectrometer(aviris). Website, http://aviris.jpl.nasa.gov/html/aviris.cube.html, 2 December 2009. 73, 77

[76] (2009), Hypercube. Website, http://www.agc.army.mil/Hypercube/, October 2009. 73, 77

[77] Rosset, A. (2003), Dicom sample image sets. Website, http://pubimage.hcuge.ch:8080/, November 2008. 80

[78] Said, A. and Pearlman, W. A. (1993) Reversible image compression via multiresolution representation and predictive coding. vol. 2094, pp. 664–674, SPIE, http://link.aip.org/link/?PSI/2094/664/1. 80

[79] Fuhr, H., Demaret, L., and Friedrich, F. (2006) *Beyond Wavelets: New Image Representation Paradigms*, chap. Chapter 7, p. 179206. CRC Press. 85

[80] Mallat, S. and Falzon, F. (1998) Analysis of low bit rate image transform coding. *Signal Processing, IEEE Transactions on*, **46**, 1027 –1042. 86, 87

[81] Pennec, E. L. and Mallat, S. (2005) Bandelet image approximation and compression. *Multiscale Modeling and Simulation*, **4**, 992–1039, http://link.aip.org/link/?MMS/4/992/1. 86

[82] Xie, G., Xie, G., Qu, X., Yan, J., and Yan, J. (2008) Bandelet image coding based on spiht. *IET Seminar Digests*, **2008**, 297–301. 88

[83] Eslami, R. and Radha, H. On low bit-rate coding using the contourlet transform. *Proc. of Asilomar Conference on Signals, Systems, and Computers*. 88

[84] Eslami, R. and Radha, H. (2004) Wavelet-based contourlet coding using an spiht-like algorithm. *Proc. of Conference on Information Sciences and Systems (CISS)*, vol. 1, pp. 784–788. 89, 90, 121

[85] Xiu-wei, T., Xi-feng, Z., and Tie-fu, D. (2008) Wavelet-based contourlet coding using speck algorithm. *Signal Processing, 2008. ICSP 2008. 9th International Conference on*, pp. 1203 –1206. 89, 90, 121

[86] Bamberger, R. and Smith, M. (1991) A multirate filter bank based approach to the detection and enhancement of linear features in images. *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, Apr., pp. 2557 –2560 vol.4. 90

[87] Park, C.-H., Lee, J.-J., Smith, M., il Park, S., and Park, K.-H. (2004) Directional filter bank-based fingerprint feature extraction and matching. *Circuits and Systems for Video Technology, IEEE Transactions on*, **14**, 74 – 85. 90, 111

[88] Bamberger, R. and Smith, M. (1990) Efficient 2-d analysis/synthesis filter banks for directional image component representation. *Circuits and Systems, 1990., IEEE International Symposium on*, May, pp. 2009 –2012 vol.3. 90

[89] Bamberger, R. and Smith, M. (1990) Narrow band analysis of a filter bank for the directional decomposition of images. *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, Apr., pp. 1739 –1742 vol.3. 90

[90] Bamberger, R. and Smith, M. (1992) A filter bank for the directional decomposition of images: theory and design. *Signal Processing, IEEE Transactions on*, **40**, 882 –893. 90

[91] Phoong, S.-M., Kim, C., Vaidyanathan, P., and Ansari, R. (1995) A new class of two-channel biorthogonal filter banks and wavelet bases. *Signal Processing, IEEE Transactions on*, **43**, 649 –665. 90, 97, 99

[92] Eslami, R. and Radha, H. (2005) New image transforms using hybrid wavelets and directional filter banks: analysis and design. *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 1, pp. I – 733–6. 90, 116

[93] Vetterli, M. and Kovačevic, J. (1995) *Wavelets and Subband Coding*. Prentice-Hall, Inc. 94

[94] Zhang, S. (2008) *The nonredundant contourlet transform (NRCT): A multiresolution and multidirection image representation with perfect reconstruction property*. Ph.D. thesis, Department of Electrical and Computer Engineering, Memorial University of Newfoundland, m Eng Thesis. 98, 111

[95] Eslami, R. (2010), Hwd matlab toolbox. Website, http://cbig.rochester.edu/people/Eslami_files/Papers/Eslami_HWD_Toolbox.zip, Access on December 2010. 106

[96] of Investigation, F. B. (1993) Wsq gray-scale fingerprint image compression specification. Tech. Rep. IAFIS-IC-0110v2, Washington, DC. 111

[97] Brislawn, C. (2002) The fbi fingerprint image compression specification. Topiwala, P. (ed.), *Wavelet Image and Video Compression*, vol. 450 of *The International Series in Engineering and Computer Science*, pp. 271–288, Springer Netherlands, 10.1007/0-306-47043-816. 111

[98] Zhang, S. and Moloney, C. (2009) Gray-scale fingerprint image compression based on the hybrid-nrct. *Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, 2009. DSP/SPE 2009. IEEE 13th*, pp. 548 –553. 111

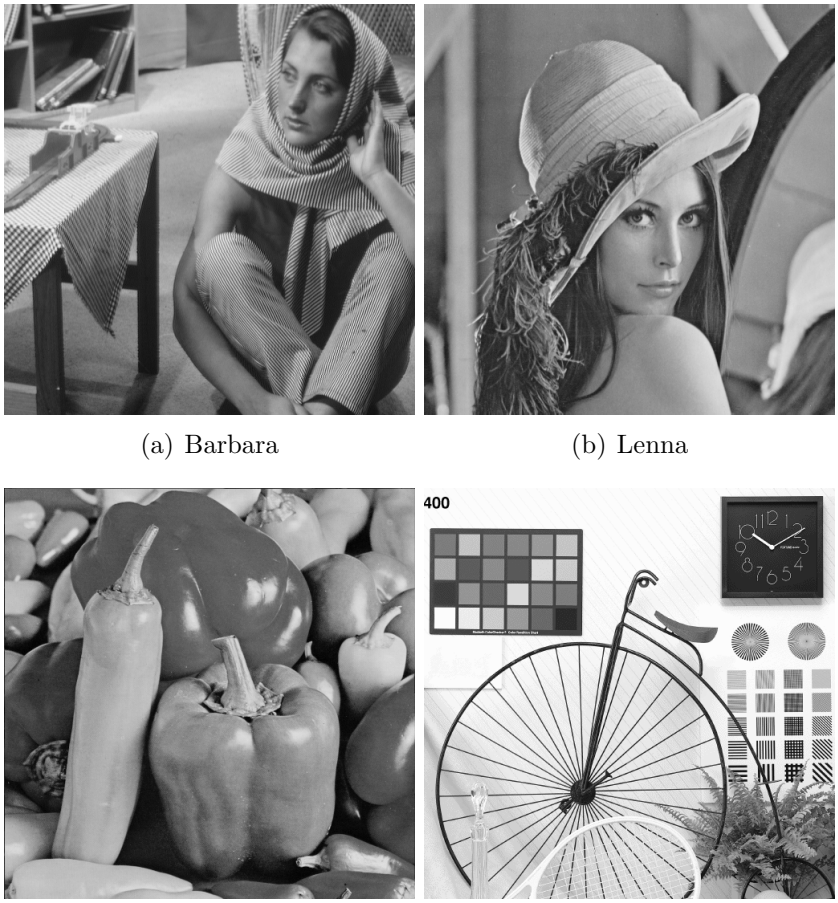
[99] of Standards, N. I. and Technology's(NIST) (2010), Nist 8-bit gray scale images of fingerprint image groups(figs). website, http://www.nist.gov/srd/nistsd4.cfm, December 2010. 111

[100] Belbachir, A. N. and Goebel, P. M. (2005) Color image compression: Early vision and the multiresolution representations. Kropatsch, W. G., Sablatnig, R., and Hanbury, A. (eds.), *Pattern Recognition*, vol. 3663 of *Lecture Notes in Computer Science*, pp. 25–32, Springer Berlin / Heidelberg, 10.1007/115505184. 117, 118

[101] Okarma, K. (2009) *Two-Dimensional Windowing in the Structural Similarity Index for the Colour Image Quality Assessment*, vol. 5702 of *Lecture Notes in Computer Science*, pp. 501–508. Springer Berlin / Heidelberg, 10.1007/978-3-642-03767-261. 126

Appendix A

Examples of test images



(a) Barbara (b) Lenna

(c) Peppers (d) Bicycle

Figure A.1: Greyscale images

Appendix B

List of publications

1. Ngadiran, R.; Boussakta, S.; Bouridane, A.; Khelifi,F.; "Exploiting chrominance planes similarity on listless quadtree coders," IET-Image Processing, -Submitted on 12 January 2012.

2. Ngadiran, R.; Boussakta, S.; Bouridane, A.; Sharif, B.; , "Hyperspectral image compression with modified 3D SPECK," Communication Systems Networks and Digital Signal Processing (CSNDSP), 2010 7th International Symposium on , vol., no., pp.806-810, 21-23 July 2010 http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5580321&isnumber=5580291

3. Ngadiran, R.; Boussakta, S.; Sharif, B.; Bouridane, A.; , "Efficient implementation of 3D listless SPECK," Computer and Communication Engineering (ICCCE), 2010 International Conference on , vol., no., pp.1-4, 11-12 May 2010 doi: 10.1109/ICCCE.2010.5556843 http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5556843&isnumber=5556742

4. Ngadiran. R, Boussakta. S., Bouridane.A., and Syarif. B.S. Low Bit Rate video compression in the 3D wavelet domain. Postgraduate Conference 2010, Newcastle University, 13 -14 January 2010